

---

Themenheft Nr. 34:

Forschung und Open Educational Resources – Eine Momentaufnahme für Europa

Herausgegeben von Markus Deimann

## Erstellung und Weiterentwicklung von Open Educational Resources im Selbstversuch

Jens Lechtenbörger

### Zusammenfassung

*Open Educational Resources (OER) versprechen einerseits den Abbau von Hürden im Bildungszugang und andererseits die Vermeidung redundanter Arbeit bei der Erstellung ähnlicher und gleichzeitig qualitativ hochwertiger Bildungsressourcen in unterschiedlichen Organisationen. Der Verbreitung von OER stehen jedoch bekannte Hürden gegenüber, wobei das ALMS-Framework einen Rahmen für die Bewertung der Wieder- und Weiternutzung von OER aus technischer Sicht bereitstellt. Ausgehend von einem Selbstversuch zur OER-Einführung werden in dieser Arbeit das ALMS-Framework erweiternde Anforderungen an OER basierend auf Konzepten aus Software-Entwicklung und technischem Schreiben definiert. Unter Beachtung dieser Anforderungen werden zwei OER-Projekte beschrieben: Zum einen wird die Weiterentwicklung eines Lehrbuchs unter Creative-Commons-Lizenz skizziert. Zum anderen werden Erstellung und Nutzung der neu entwickelten Software emacs-reveal für die Erzeugung von für das Selbststudium geeigneten, mit Audiokommentaren unterlegten HTML-Präsentationen beschrieben; die Präsentationen werden in einfachen Textdateien erstellt, wobei die Erzeugung von HTML-Code automatisiert in einer öffentlichen GitLab-Infrastruktur abläuft und damit die Software-Nutzung vereinfacht. Ergebnisse einer Umfrage unter Studierenden verdeutlichen die Vorzüge der erzeugten Präsentationen.*

### Creation and refinement of Open Educational Resources in a self-experiment

#### Abstract

*Open Educational Resources (OER) promise to reduce entry barriers towards educational resources as well as to avoid redundant work when similar educational resources of high quality are created in different organizations. However, adoption of OER faces known challenges, and the ALMS framework provides evaluation criteria for the re-use of OER from a technical perspective. Starting from a self-experiment on OER adoption, novel requirements for OER based on the ALMS framework are defined by taking concepts from software engineering and technical writing into account. Respecting these requirements, two OER projects are reported: First, the creation of a new edition of a textbook under a Creative Commons license is sketched. Second, the development and use of a new software*

*tool, called emacs-reveal, is described. This tool allows to generate HTML presentations with embedded audio explanations, which are suitable for self-study. The generation of HTML from simple text files takes place in a public GitLab infrastructure, increasing the software's ease of use. Students' responses to a questionnaire demonstrate advantages of generated presentations.*

## Einleitung

Die seit einer UNESCO-Initiative im Jahre 2002 propagierten Open Educational Resources (OER), zu Deutsch offene Bildungsressourcen, versprechen einerseits den Abbau von Hürden im Bildungszugang und andererseits die Reduktion redundanter Arbeit bei der Erstellung ähnlicher und gleichzeitig qualitativ hochwertiger Bildungsressourcen in unterschiedlichen Organisationen. Obwohl seit dem Jahre 2011 im Rahmen von Open Educational Practices die Notwendigkeit eines Perspektivwechsels von Ressourcen zu Praktiken beworben worden ist (vgl. Andrade et al. 2011; Ehlers 2011), beobachtet der Autor als Universitätsdozent in seinem Umfeld kein nennenswertes Bewusstsein für OER, von routinemässigem Praxiseinsatz ganz zu schweigen. In diesem Beitrag berichtet er aus bildungstechnologischer Perspektive von Projektergebnissen eines Selbstversuchs zu OER im Sommersemester 2017, der die Erstellung von OER unter folgender Fragestellung adressierte: Welche technischen Anforderungen bestehen einerseits an OER, die sich für das Selbststudium im Rahmen der Lehr- und Lernstrategie Just-in-Time-Teaching (JiTT, vgl. Novak et al. 1999; McGee, Stokes, und Nadolsky 2016) eignen, und andererseits an Werkzeuge und Methoden zur OER-Entwicklung, die eine gemeinschaftliche OER-Weiterentwicklung anstreben?

OER sind nach Atkins, Brown und Hammond (2007) Lehr-, Lern- und Forschungsressourcen, die gemeinfrei (engl. *public domain*) oder so lizenziert sind, dass sie von anderen frei (wieder- und weiter-) genutzt werden können; typischerweise kommen diverse Creative-Commons- (CC-) Varianten als Lizenzen für OER zum Einsatz (vgl. Hilton et al. (2010) sowie Mruck et al. 2013). Neben lizenzrechtlichen Fragestellungen wurden im Laufe der Zeit zahlreiche Hürden identifiziert, welche der Verbreitung von OER im Wege stehen (vgl. Andrade et al. 2011; Jung, Sasaki, und Latchem 2016). Der *Ljubljana OER Action Plan 2017*<sup>1</sup> formuliert Aktionen zur Überwindung derartiger Hürden und gibt als erstes Ziel «den Aufbau der Fähigkeit von Benutzerinnen und Benutzern zum Finden, Weiterverwenden, Erzeugen und Teilen von OER» aus. Eine der diesem Ziel dienenden Aktionskategorien lautet «Aufbau von Bewusstsein und Fähigkeiten für die OER-Nutzung», was durch zehn OER-Aktionen weiter präzisiert wird, darunter OER-Aktion (c):

---

1 [https://en.unesco.org/sites/default/files/ljubljana\\_oer\\_action\\_plan\\_2017.pdf](https://en.unesco.org/sites/default/files/ljubljana_oer_action_plan_2017.pdf)

«Disseminate the findings of research on OER to support models of good practice with a focus on cost-effectiveness, sustainability, exploration of new tools and technologies for the creation and sharing of OER;»

Im Einklang mit den Zielen dieser Aktion beinhaltet der vorliegende Artikel ausgehend von einer im folgenden Abschnitt skizzierten Lehrsituation ohne OER folgende Beiträge:

- Zunächst werden technische Anforderungen an OER in Kombination des sogenannten ALMS-Frameworks von Hilton et al. (2010) als Rahmen zur Bewertung der Wieder- und Weiternutzbarkeit von OER mit Ansätzen der Software-Entwicklung (gemeinschaftliche Software-Erstellung als Freie/Libre und Open Source Software, vgl. Feller et al. (2005), und verteilte Versionsverwaltung, beispielsweise mit Git, vgl. Chacon und Straub 2014) sowie des technischen Schreibens (Single Sourcing nach Rockley 2001) definiert.
- Die resultierenden Anforderungen erlauben mit Blick auf oben zitierte OER-Aktion (c) eine über bisherige Ansätze hinausgehende Bewertung von Werkzeugen und Technologien für Erzeugung und Austausch von OER. Insofern leisten sie aus bildungstechnologischer Perspektive einen Beitrag zur Definition wünschenswerter Eigenschaften von OER und OER-Werkzeugen, die von Software-Entwicklern, Lehrenden und Lernenden angestrebt werden können.
- In Abschnitt «Entwickelte Inhalte und Software» wird der im Semesterverlauf vollzogene Übergang zu OER unter Einhaltung der zuvor entwickelten Anforderungen dargestellt, und zwar zunächst am Beispiel der Weiterentwicklung eines CC-lizenzierten Lehrbuchs.
- Die Beschreibung verfolgt das Ziel, ein Bewusstsein für mögliche OER-Ansätze zu schaffen, und lässt sich als Werbung für eine von Hefferon und Schueller (2016) diskutierte Good Practice mit niedrigen Einstiegshürden auffassen.
- Im Anschluss wird mit *emacs-reveal* eine neu entwickelte Software zur Erstellung von für das Selbststudium geeigneten, mit Audiokomentaren unterlegten HTML-Präsentationen präsentiert.
- Diese Software ist in die öffentliche GitLab-Infrastruktur eingebettet, erfüllt die zuvor definierten Anforderungen und verkörpert ein Werkzeug für Erstellung und Austausch von OER im Sinne von OER-Aktion (c).

Erste Umfrageergebnisse verdeutlichen in Abschnitt «Erste Umfrageergebnisse», dass ein Grossteil der Studierenden den Entscheidungen des Lehrenden zustimmt, bevor der Artikel mit Zusammenfassung und Ausblick in Abschnitt «Zusammenfassung und Ausblick» endet.

## Ausgangssituation

### *Hintergrund*

Der Autor hält die Vorlesung *Operating Systems* im englischsprachigen Pflichtmodul *Computer Structures and Operating Systems* des Bachelor-Studiengangs Wirtschaftsinformatik an der WWU Münster. Der Vorlesung lag ein nicht offenes Lehrbuch mit einem vom Verlag für Dozenten zur Verfügung gestellten, nicht offenen Foliensatz (Microsoft Powerpoint) zugrunde. Über die Jahre hat der Autor diesen Foliensatz verbessert und ergänzt, aber aufgrund einschränkender Lizenzen «natürlich» nur für sich selbst und seine Studierenden, also für einen «bestimmt abgegrenzten Kreis von Unterrichtsteilnehmern» im Sinne der damals geltenden Regelungen nach § 52a UrhG, wodurch auch die Übernahme von Teilen aus ansonsten urheberrechtlich verschlossenen Forschungsarbeiten möglich ist.<sup>2</sup> Insbesondere vor dem Hintergrund, dass das Thema weltweit an zahllosen Hochschulen in informatikbezogenen Studiengängen gelehrt wird, verkörpert die Erstellung zugehöriger Lehrmaterialien für die ausschliessliche Nutzung an einer Hochschule eine Vergeudung von Expertise, Zeit und Ressourcen.

*Operating Systems* fand in den Vorjahren als klassische Vorlesung statt, wobei der 90-minütige Vorlesungsrhythmus durch interaktive Aufgaben und Online-Umfragen aufgelockert wurde. Im Zuge der durch ein «Fellowship für Innovationen in der digitalen Hochschullehre»<sup>3</sup> des Ministeriums für Innovation, Wissenschaft und Forschung des Landes Nordrhein-Westfalen und des Stifterverbandes geförderten Umstellung der Vorlesung auf Just-in-Time-Teaching (JiTT, vgl. Novak et al. 1999; McGee, Stokes, und Nadolsky 2016) entschied sich der Autor, für die neu zu konzipierende Vorlesung im Sommersemester 2017 sämtliche Lehr- und Lernmaterialien von Grund auf neu zu erstellen und als OER zu publizieren.

Das Konzept von JiTT besteht unabhängig vom Einsatz von OER und sieht vor, dass sich Studierende Themen im Selbststudium erschliessen und Aufgaben (engl. *warm-ups*) zu diesen Themen bearbeiten. Die Lösungen dieser Aufgaben werden kurzfristig korrigiert, wodurch falsche Vorstellungen und Missverständnisse aufgedeckt werden können, die im Rahmen klassischer Vorlesungen unentdeckt bleiben und dort dem weiteren Lernfortschritt entgegen stehen. Basierend auf den Erkenntnissen der Warm-Up-Korrekturen werden im Zuge von JiTT die Präsenzveranstaltungen *just in time* angepasst, um Missverständnisse unmittelbar aufzuklären. Für weitergehende Informationen zu JiTT sei auf das Verbundprojekt HD MINT verwiesen (vgl. Zentrum für Hochschuldidaktik (DiZ) 2016), in dem über mehrere Jahre zahlreiche Lehrveranstaltungen bayerischer Hochschulen für angewandte Wissenschaften auf JiTT umgestellt wurden.

---

2 Seit dem 1. März 2018 gilt in Deutschland § 60a UrhG.

3 <https://www.stifterverband.org/lehrfellows/2016/lechtenboerger>

## **Anforderungen**

Um das Selbststudium im Rahmen von JiTT zu unterstützen, bedarf es geeigneter Materialien, für welche die Methode keine Einschränkungen definiert. Naheliegender erscheint zunächst der Einsatz von Videos, die nach Bishop und Verleger (2013) auch im Flipped Classroom zum Einsatz kommen. Der Autor entschied sich jedoch aufgrund (a) des erwarteten Erstellungsaufwandes aus Sicht eines Lehrenden, (b) der vergleichsweise eingeschränkten Navigation und Suche in Videos aus Perspektive eines Lernenden und (c) nachfolgend diskutierter objektiver Anforderungen für die Erstellung von Präsentationen, die Audiokommentare beinhalten sollten.

Für die Auswahl von Software mit zur Erstellung derartiger Präsentationen geeigneten Abläufen können Best Practices und Anforderungen unterschiedlicher Disziplinen herangezogen werden; naheliegend sind natürlich diejenigen, die in Publikationen zu OER erarbeitet wurden. Darüber hinaus werden im Folgenden Konzepte von Software-Entwicklung von Feller et al. (2005) dargestellte «Freie Software» und verteilte Versionsverwaltung mit Git (vgl. Chacon und Straub 2014) sowie «Single Sourcing» nach Rockley (2001) aus dem Umfeld des technischen Schreibens in ihrer Bedeutung für OER diskutiert.

Gemeinsam mit den vier «Rs» (*reuse, redistribute, revise, remix*; später von Wiley (2014) um *retain* zu fünf «Rs» erweitert) zur Charakterisierung von Offenheit wird von Hilton et al. (2010) das ALMS-Framework mit seinen Kategorien

- «A» (Access to editing tools),
- «L» (Level of expertise required to revise or remix),
- «M» (Meaningfully editable) und
- «S» (Source-file access)

zur Präzisierung technischer Offenheit vorgestellt. Diese Kategorien werden nachfolgend präsentiert, ergänzt und als resultierende Anforderungen an OER in Tabelle 1 zusammengefasst.

Kategorie «A» befasst sich mit der zur Bearbeitung einer OER notwendigen Software, die (a) Einschränkungen hinsichtlich vorausgesetzter Betriebssysteme unterliegen kann oder auch nicht und die (b) als freie oder als proprietäre Software lizenziert sein kann. Kategorie «L» bezieht sich auf die zur Software-Nutzung benötigten Expertise, die möglichst gering sein sollte. Kategorie «M» fordert die Editierbarkeit der erstellten Materialien, Kategorie «S» die Verfügbarkeit der Quellmaterialien.

Die Kategorien «M» und «S» sind eng miteinander verknüpft: Der CC-lizenzierte Scan einer Zeichnung rangiert am unteren Ende der Kategorie «M», was im Widerspruch zu Kategorie «S» daraus resultiert, dass die (analoge) Quelle des Scan-Ergebnisses nicht vorliegt. Im Falle einer CC-lizenzierten Office-Präsentation, die etwa mit LibreOffice Impress oder Microsoft Powerpoint erstellt wurde, können PDF-Versionen oder Originalpräsentationen weitergegeben werden. Offenbar ist die Präsentation nur dann vernünftig editierbar (Kategorie «M») und damit weiterverwendbar, wenn

die Originalpräsentation verfügbar ist (Kategorie «S»), aber nicht wenn lediglich aus der Software generierte PDF-Dateien vorliegen. Verallgemeinernd lässt sich feststellen, dass die Verfügbarkeit der Quellmaterialien («S») eine notwendige Voraussetzung für sinnvolle Bearbeitbarkeit («M») ist. Im Falle von Videos sei jedoch erwähnt, dass dem Autor die Bearbeitbarkeit von Videos aus fremden Quellen generell eingeschränkt erscheint: Während das Schneiden relativ einfach möglich ist, übersteigen die Anforderungen zur Korrektur von Inhalten oder zum Einfügen neuer Abschnitte mit weitergehenden Erläuterungen unter Erhalt des ursprünglichen Look-and-Feel seine Fähigkeiten.

Über die vier ALMS-Kategorien hinaus werden in dieser Arbeit weitergehende technische Anforderungen definiert, die Nutzung, Wieder- und Weiterverwendung vereinfachen: Wie oben erwähnt umfasst die Kategorie «A» des ALMS-Frameworks bezüglich der Erstellung von OER sowohl (a) Betriebssystemunabhängigkeit als auch (b) Freiheit der Bearbeitungssoftware. Bezüglich (b) sei ausdrücklich betont, dass der Mitte der 1980er Jahre von Stallman (1986) geprägte Begriff der *freien Software* eine klar definierte Bedeutung hat und durch Lizenzen zu gewährende Freiheiten der Nutzerinnen und Nutzer hinsichtlich der Ausführung, Änderung und Weitergabe von Software anstrebt (und nicht notwendigerweise deren kostenlose Verteilung), was in seiner Bedeutung im OER-Umfeld von Vukovic et al. (2009) ausgiebig diskutiert wird. Kurz zusammengefasst gewährt freie Software folgende vier Freiheiten<sup>4</sup>, die deutliche Parallelen zu den fünf Rs aufweisen:

- Die Freiheit, das Programm zu jedem Zweck auszuführen.
- Die Freiheit, die Funktionsweise der Software zu untersuchen und anzupassen.
- Die Freiheit, die Software weiterzugeben.
- Die Freiheit, die Software zu verbessern und verbessert weiterzugeben.

In dieser Arbeit wird freie Software mit dem auch von Vukovic et al. (2009) erwähnten Akronym FLOSS für Freie/Libre und Open Source Software bezeichnet. Für den hier dargestellten Selbstversuch wurde ausschliesslich FLOSS verwendet.

Bezüglich der von Kategorie «A» angestrebten Betriebssystemunabhängigkeit lässt sich ergänzend und verallgemeinernd festhalten, dass neben der betriebssystemunabhängigen *Bearbeitung* von OER auch ihre plattformunabhängige *Erarbeitung* wünschenswert ist: Grösstmögliche Nutzbarkeit von OER ergibt sich offenbar durch die *plattformunabhängige* Darstellung, Wiedergabe und Erarbeitung von OER mit FLOSS. Lernende sollten OER mit einer möglichst grossen Bandbreite von Geräten erarbeiten können, insbesondere auf mobilen Geräten und auch dann, wenn keine Internetverbindung vorliegt. Dies gestaltet sich beispielsweise für Powerpoint und LibreOffice auf mobilen Geräten nach Thorogood (2016) als schwierig und schliesst z.B. mit Beamer-LaTeX erstellte PDF-Dateien mit eingebetteten Audiodateien aus,

---

4 <https://www.gnu.org/philosophy/free-sw.de.html>

deren Wiedergabe auf vielen Geräten scheitert. Generell bestehen für PDF-Dateien zwar vielfältige Gestaltungsmöglichkeiten (Einbettung von Medien, Formularen, aktiven Inhalten), jedoch werden diese nach Glushko (2015) auf unterschiedlichen Zielsystemen nicht einheitlich umgesetzt oder gar nicht unterstützt (vgl. Shaffer, Naps, und Fouh 2011). Demgegenüber hat sich der Web-Browser auf nahezu allen Geräten und Plattformen (wie Desktop-PCs, Tablets, Smartphones mit unterschiedlichen Betriebssystemen) als Zugang zu unterschiedlichsten Anwendungen etabliert, sodass auch interaktive und multimediale Lehrbücher auf HTML5-Basis erstellt werden können (vgl. Shaffer, Naps, und Fouh 2011).

Das aus dem Umfeld des technischen Schreibens stammende, von Rockley (2001) definierte Konzept des *Single Sourcing* kann als Ergänzung und Verallgemeinerung der Kategorien «M» und «S» aufgefasst werden: Single Sourcing wurde im XML-Umfeld entwickelt und beruht auf der für XML-Dokumente typischen Trennung von Inhalt und Layout: Autoren legen die Dokumentstruktur (etwa in Kapiteln und Abschnitten) fest und konzentrieren sich dann auf Inhalte (nicht nur Text, sondern beispielsweise auch Abbildungen). Wie Überschriften unterschiedlicher Gliederungsebenen dann formatiert oder an welchen Stellen Abbildungen gezeigt werden, um ein ansprechendes Layout zu erzielen (das auf verschiedenen Geräten unterschiedlich gestaltet sein muss), wird separat definiert, womöglich von Layout-Experten. Single Sourcing verinnerlicht diese Trennung von Inhalt und Layout und verkörpert eine extreme Form der Wiederverwendung von Texten aus einer einzigen Quelle. Es setzt auf die Wieder- und Weiternutzung eines Dokuments in unterschiedlichen Kontexten und an verschiedenen Stellen, ohne Copy&Paste zu nutzen. Single Sourcing strebt dabei die Darstellung auf verschiedenen Geräten und Plattformen und in unterschiedlichen Medienformaten (wie HTML und PDF) an. Glushko (2015) berichtet beispielsweise von der Erstellung eines transdisziplinären Lehrbuches unter Nutzung von Single Sourcing auf Basis von XML, das als Familie von Büchern aufgefasst werden kann, die aus einer Quelle in Varianten für unterschiedliche Adressatenkreise sowohl gedruckt als auch in verschiedenen E-Book-Formaten ausgeliefert werden können.

Es erscheint naheliegend, das Single-Sourcing-Prinzip auch im Zuge der Weiterentwicklung von OER einzusetzen. In Ergänzung der Kategorien «M» und «S» des ALMS-Frameworks erfordert eine organisationsübergreifende Zusammenarbeit (mit unterschiedlichen Layout-Vorgaben, etwa hinsichtlich Corporate Identity) an OER nicht nur, dass Quelldateien verfügbar und editierbar sind, sondern auch, dass sie unter *Trennung von Inhalt und Layout* bearbeitet werden können und dass es eine Quelle gibt, die gemeinschaftlich weiterentwickelt wird. Zu beachten ist, dass gängige Präsentationssoftware wie LibreOffice und Powerpoint mit dem Konzept des sogenannten Folien-Masters zwar einen Schritt zur Trennung von Inhalt und Layout beinhaltet, dieser in der Praxis aber nicht ausreicht: Nach Erfahrung des Autors erfordert die Umstellung des Folien-Masters an universitätsinterne Designvorgaben einen

immensen Aufwand, da Positionierungen von Texten und Grafiken regelmässig zerstört werden (durch andere Schriftarten und Grössenvorgaben).

Im Hinblick auf Single Sourcing ist es zudem hilfreich, wenn die Quelldateien ein Format aufweisen, das die Nachverfolgung und Integration von Korrekturen oder Ergänzungen ermöglicht. Auch dieses Kriterium wird von gängiger Präsentationssoftware nicht erfüllt: Wenn Lehrende bei Semesterende «ihre» Versionen an den ursprünglichen Autor zurückschicken würden, müsste letzterer die neuen Versionen in mühsamer Kleinstarbeit durchgehen, um inhaltliche Änderungen zu identifizieren und in das Original einzuarbeiten. In Ergänzung der Kategorie «S» ermöglicht ein *einfaches Format* der Quelldateien dagegen die Anwendung von in der Softwareentwicklung üblichen Kooperationswerkzeugen durch Versionsvergleiche und -integrationen auch für OER. So plädieren Hefferon und Schueller (2016) im Rahmen der Erstellung offener Lehrbücher für die Verwendung von LaTeX-Quelldokumenten, die mit Versionsverwaltungssystemen wie Git (vgl. Chacon und Straub 2014) auf Web-Plattformen wie GitHub<sup>5</sup> oder GitLab<sup>6</sup> gepflegt werden können, was im folgenden Abschnitt exemplarisch illustriert wird. Essentiell ist hier, dass Git ein *dezentralisiertes* Versionsverwaltungssystem repräsentiert, bei dem unterschiedliche Autoren an unterschiedlichen Orten mit unterschiedlichen Versionen gemeinsamer Dokumente arbeiten können, wobei die Bearbeitung der Dokumente *keine* Internet-Verbindung voraussetzt, also auch offline erfolgen kann. Lediglich für den Austausch (mit nachfolgendem Abgleich und Zusammenführen) von Versionen wird eine Internet-Verbindung benötigt.

Es sei betont, dass die im folgenden Abschnitt beschriebene FLOSS *emacs-reveal* sowie die damit erstellten OER-Präsentationen bewusst auf GitLab publiziert sind (statt des populäreren GitHub). Da GitLab selbst als FLOSS entwickelt wird, beruht der vorgestellte Ansatz durchgängig auf FLOSS, insbesondere auch für die Bereitstellung von OER, die damit sowohl in der öffentlichen GitLab-Infrastruktur erfolgen kann als auch auf selbst betriebenen Servern. Diese Wahlfreiheit ist charakteristisch für von Kerres und Heinen (2014) diskutierte *offene* informationelle Ökosysteme zur Bereitstellung, Suche und Erarbeitung von OER: Offene Systeme vermeiden den «Vendor Lock-In» abgeschlossener Systeme, der problematisch für die Wissensgesellschaft sei, die von freier und gemeinschaftlicher Entwicklung von und freiem Zugang zu Wissen abhängig sei.

---

5 <https://github.com/>

6 <https://about.gitlab.com/>



Ziel	Anforderung	Beispiel
ALMS	Access to editing tools	FLOSS (z.B. LibreOffice, LaTeX, Textformate)
	Level of expertise required to revise or remix	WYSIWYG <sup>7</sup> oder Textformate
	Meaningfully editable	LaTeX, Org Mode <sup>8</sup> , HTML
	Source-file access	LaTeX, Org Mode, HTML
Plattformunabhängigkeit	FLOSS für OER-Erstellung, auch offline	LibreOffice, LaTeX, Textformate
	FLOSS für OER-Bereitstellung	GitLab, Wiki
	FLOSS für OER-Nutzung	LibreOffice, PDF, HTML
	Mobile OER-Nutzung, auch offline	HTML, PDF eingeschränkt <sup>9</sup>
Single Sourcing	Eine Quelle ohne Copy&Paste	Git-Repository, Wiki
	Trennung von Layout und Inhalt	LaTeX, (X)HTML + CSS
	Einfaches Textformat	LaTeX, Org Mode

**Tab. 1.:** Anforderungen an OER-Software.

Die bisher diskutierten Anforderungen sind in Tabelle 1 zusammengefasst und werden im weiteren Verlauf zur Bewertung eingesetzter und entwickelter Techniken herangezogen.

## Entwickelte Inhalte und Software

### Übergang zu OER-Lehrbuch

Eine Suche nach englischsprachigen OER-Lehrbüchern zum Thema *Operating Systems* lieferte mehrere Kandidaten, von denen der Autor aufgrund grosser Überschneidungen zu bisherigen Vorlesungsinhalten das Werk Hailperin (2016) auswählte, das unter CC-Lizenz verfügbar ist und dessen LaTeX-Quelltext als Projekt auf der Web-Plattform GitHub verfügbar ist, womit die Empfehlungen von Hefferon und Schueller (2016) für offene Lehrbücher sowie sämtliche Anforderungen aus Tabelle 1 erfüllt werden.

Ergänzungsvorschläge erarbeitete der Autor zunächst auf einer lokalen Kopie, einem sog. *Fork* (vgl. Chacon und Straub 2014). Während das Erzeugen der Kopie im Wesentlichen einem Download entspricht, konnte die inhaltliche Arbeit offline stattfinden, bevor die Ergänzungen wiederum zu GitHub synchronisiert und dort über einen sog. *pull request* dem Lehrbuchautor Hailperin präsentiert wurden, der

7 WYSIWYG = What you see is what you get; Bearbeitung in einer grafischen Umgebung, deren Bildschirm-darstellung das Endergebnis zeigt.

8 <https://orgmode.org/>

9 Multimediaeinbettung nicht plattformunabhängig, s. Text.

sie annahm<sup>10</sup>, was rechtzeitig vor Vorlesungsbeginn zur Neuauflage Hailperin (2017) führte. Abbildung 1 zeigt einen Ausschnitt des Browser-Fensters zu Änderungen an der Quelldatei `persistence.tex`, der die Einfachheit kollaborativer OER-Erstellung in Git-Plattformen verdeutlicht: Im unteren Bereich werden Quelltextzeilen mit Zeilennummern dargestellt. Gelöschten Zeilen geht ein Minuszeichen voran, und sie sind rot markiert; eingefügten Zeilen ist bei grüner Färbung ein Pluszeichen vorangestellt. Diese Ansicht erlaubt es dem Lehrbuchautor, vorgeschlagene Änderungen schnell zu erfassen (um sie dann zu akzeptieren oder auch nicht). Oberhalb des Quelltextes ist zudem zu sehen, dass eine Unterhaltung (Conversation) zu den Änderungen stattgefunden hat und dass insgesamt drei Quelldateien betroffen waren. Zur Vereinfachung der Zusammenarbeit können sich Projektbeteiligte per E-Mail über verschiedene Ereignisse informieren lassen.

Mit Bezug auf das Single-Sourcing-Prinzip lässt sich Folgendes feststellen. Da im vorliegenden Fall sämtliche Änderungsvorschläge angenommen wurden, bestand nie die Notwendigkeit, eine geänderte Version mit Copy&Paste zu erstellen. Dieser Idealfall muss natürlich nicht immer eintreffen. Bei divergierenden Auffassungen könnte der Autor seine eigene Version (als Fork) pflegen, die mit Git-Mechanismen so mit dem Original verknüpft bleibt, dass sich Weiterentwicklungen des Originals *ohne* Copy&Paste immer noch in der eigenen Version niederschlagen – über Mechanismen wie `merge` und `rebase` (vgl. Chacon und Straub 2014). Entsprechend unterstützt Git selbst im Fall divergierender Versionen einer OER das Single-Sourcing-Prinzip. Voraussetzung ist dafür lediglich, dass die Quelldateien in einem Textformat vorliegen.

### ***Erstellung und Nutzung von emacs-reveal***

Zur Erstellung von Präsentationen mit erläuternden Audiokommentaren verlief die Suche nach geeigneter Software, die den Anforderungen aus Tabelle 1 genügt, ergebnislos. Klassische Präsentationssoftware wie LibreOffice wurde verworfen, da diese keine Kollaboration nach dem Single-Sourcing-Prinzip erlaubt (kein einfaches Textformat, keine Trennung von Layout und Inhalt). Bei Experimenten mit Beamer-LaTeX<sup>11</sup> konnten zwar PDF-Präsentationen mit eingebetteten Audiodateien erzeugt werden, jedoch mit FLOSS nicht plattformübergreifend wiedergegeben werden, was sich mit den Erfahrungen von Shaffer, Naps, und Fouh (2011) deckt.

---

10 <https://github.com/Max-Hailperin/Operating-Systems-and-Middleware--Supporting-Controlled-Interaction/issues/125>

11 <https://github.com/josephwright/beamer/>

The screenshot shows a GitHub pull request interface. At the top, the repository name is 'Max-Hailperin/Operating-Systems-and-Middleware--Supporting-Controlled-Interaction'. Below that, there are navigation tabs for 'Code', 'Issues (3)', 'Pull requests (0)', 'Projects (0)', and 'Insights'. The main title of the pull request is 'Use binary prefixes for main memory sizes (#122)'. It was authored by 'lechten' and committed on 'Jun 7, 2017'. The commit message is '1 parent 33d6fc6 commit 0a74fe26db84'. Below the commit information, it says 'Showing 3 changed files with 70 additions and 58 deletions'. A diff view for the file 'persistence.tex' is shown, with line numbers 792 to 800. The diff highlights changes from kilobytes to kibibytes in several lines.

Line	Change	Text
792		virtual addresses within address spaces to physical addresses within
793		memory. In a file system, the mapping is from positions within files
794		to locations in persistent storage. For efficiency, the mapping is done at a coarse
795	-	granularity, several kilobytes at a time. In virtual memory, each
795	+	granularity, several kibibytes at a time. In virtual memory, each
796		page is mapped into a page frame; in a file system, each block of a
797	-	file is mapped into a storage block. (You will see that blocks are
798	-	typically several kilobytes in size, spanning multiple sectors.)
797	+	file is mapped into a storage block. (Recall from Section~ \ref{vm-reps}
798	+	that ``kibi'' is the binary unit for 1024. You will see that blocks are
799	+	typically several kibibytes in size, spanning multiple sectors.
800	+	Note, however, that transfer speeds and \emph{sizes} of hard disks are

Abb. 1.: Ansicht von Änderungen an Hailperin (2016) auf GitHub.

Thorogood (2016) stellt mit slideDeck.js ein nicht öffentlich verfügbares Werkzeug vor, mit dem im Browser anzusehendes, interaktives HTML-Lehrmaterial aus Textdateien in einer eigens entwickelten Markup-Sprache erstellt werden kann. Vorteile von HTML liegen laut Thorogood (2016) in der weiten Verbreitung von Browsern, dem einfachen Einbetten interaktiver Inhalte und den Verweismöglichkeiten zu externen Inhalten in einer einheitlichen Umgebung (im Vergleich zu PDF, wo das Folgen eines Verweises zum Start einer anderen Anwendung, des Browsers, führt). Zudem sei zum einen der Zeitaufwand zum Erstellen von HTML-Präsentationen dank der im Vergleich zu HTML einfachen Markup-Sprache erheblich reduziert. Zum anderen können sich Lehrende dank der Trennung von Layout und Inhalt auf die Inhalte konzentrieren. Obwohl Thorogood (2016) dies nicht erwähnt, könnte der Ansatz insgesamt dem Single-Sourcing-Prinzip folgen, wenn die Quelldateien beispielsweise in einem Git-Repository veröffentlicht würden.

Von den Vorteilen von HTML-Präsentationen unter Beachtung des Single-Sourcing-Prinzips überzeugt entschied sich der Autor (ohne Evaluation), das als FLOSS entwickelte HTML-Framework reveal.js<sup>12</sup> für Präsentationen einzusetzen, das in

12 <https://revealjs.com/>

Kombination von HTML5, CSS und JavaScript zahlreiche gängige Präsentationsmöglichkeiten unterstützt, z.B. eine Präsentationsansicht mit Vorschau und Notizen, Animationseffekte, Folienübergänge, Einbettung von Bildern, Tonspuren, Videos und mathematischen Formeln, responsives Design mit Touch-Unterstützung, sog. Themen für unterschiedliche Layout-Gestaltungen. Um Präsentationen in einer einfachen Markup-Sprache (statt direkt in HTML) schreiben zu können, fiel die Wahl auf das von Schulte und Davison (2011) beschriebene Org Mode<sup>13</sup>, welches der Autor tagtäglich für das Schreiben von Texten aller Art (mit umfangreichen Exportmöglichkeiten, z.B. zu LaTeX, HTML und OpenDocument-Formaten) sowie die Termin- und Aufgabenverwaltung nutzt.<sup>14</sup> Für Org Mode existiert mit Org-Reveal<sup>15</sup> eine FLOSS, um HTML-Präsentationen aus Org-Mode-Textdateien zu erzeugen (deren Möglichkeiten diejenigen von slideDeck.js umfassen). Da Org-Reveal aktuell nicht weiterentwickelt wird, für die Erstellung von Präsentationen mit Audiokommentaren jedoch Erweiterungen nötig sind, entwickelte und veröffentlichte der Autor *emacs-reveal*<sup>16</sup> als FLOSS. *Emacs-reveal* beinhaltet eine weiterentwickelte Version von Org-Reveal sowie einige Plugins für reveal.js, insbesondere für Audioinhalte.

Mit *emacs-reveal* erstellte der Autor semesterbegleitend HTML-Präsentationen für *Operating Systems*, die als GitLab Pages<sup>17</sup> verfügbar sind, aber auch heruntergeladen oder alternativ lokal erzeugt werden können und dann im Browser mobil und offline nutzbar sind. Die Org-Mode-Quelltexte sind auf GitLab<sup>18</sup> veröffentlicht. Bei Änderungen an den Quelltexten werden auf GitLab über ein eigens entwickeltes sog. Docker-Image (vgl. Merkel 2014) automatisch entsprechend geänderte Präsentationen als GitLab Pages publiziert, sodass weder für die Erzeugung der Präsentationen noch für deren Veröffentlichung im Web besondere technische Expertise nötig ist, was eine Vereinfachung im Sinne der Kategorie «L» des ALMS-Frameworks bewirkt, aber immer noch Git-Kenntnisse voraussetzt (die unter Software-Entwicklern und in informatiknahen Studiengängen weit verbreitet sind, aber ansonsten eine entsprechende Einarbeitung erfordern). Der Betrieb der skizzierten Infrastruktur auf eigenen Servern ist möglich, aber dank GitLab nicht notwendig.

Die kollaborative, dem Single-Sourcing-Prinzip folgende Weiterentwicklung der Präsentationen wird durch übliche Git-Abläufe mit den in Abbildung 1 für GitHub gezeigten Funktionen auch auf GitLab unterstützt. Zusammenfassend erfüllt das Vorgehen sämtliche der in Tabelle 1 aufgelisteten Anforderungen.

---

13 <http://orgmode.org/>

14 Org-Quelltext einer Vorversion dieses Artikels: <https://gitlab.com/lechten/publications/blob/master/oer-selbstversuch/oer-selbstversuch-lang.org>

15 <https://github.com/yjwen/org-reveal>

16 <https://gitlab.com/oer/emacs-reveal>

17 <https://oer.gitlab.io/OS/>

18 <https://gitlab.com/oer/OS>

### Erste Umfrageergebnisse

Nach der Hälfte der Vorlesungszeit zu *Operating Systems* wurde eine Online-Umfrage zur JiTT-Umstellung durchgeführt, in deren Rahmen auch Meinungen zu den mit *emacs-reveal* neu erstellten HTML-Präsentationen abgefragt wurden. Die hier relevanten Ergebnisse sind in Tabelle 2 zusammengefasst und werden nachfolgend diskutiert.

Die erste Tabellenzeile verdeutlicht, dass sich mehr als zwei Drittel der Studierenden PDF-Versionen der Präsentationen wünschten. Da die Inhalte der Präsentationen kompakt in Org-Textdateien vorliegen, die in jedem Texteditor bzw. in jeder Textverarbeitungssoftware gelesen und ergänzt werden können, hatte der Autor dieses Ergebnis nicht erwartet, allerdings erscheint es angesichts langjähriger Gewohnheiten auf Seiten der Studierenden auch nicht besonders überraschend. Aus technischer Sicht gestaltete es sich nicht schwierig, diesen Studierendenwunsch zu erfüllen: Einerseits unterstützt das zugrunde liegende HTML-Framework *reveal.js* einen PDF-Export, bei dem jede Folie als separate Seite im PDF-Dokument abgebildet wird, andererseits erweiterte der Autor die Docker-Umgebung auf GitLab um die Erzeugung von PDF-Dateien aus den Org-Quelltexten; letztere geben die Folieninhalte platzsparend ohne künstliche Seitenumbrüche wieder und sind neben den HTML-Präsentationen auf GitLab veröffentlicht.

Aussage	Teilnehmende
PDF-Versionen der Präsentationen sind notwendig.	68,52%
Ich bevorzuge HTML-Präsentationen gegenüber Videos.	42,59%
Ich bevorzuge Videos gegenüber HTML-Präsentationen.	37,04%
Ich wünsche mir weitere Audioinhalte.	40,74%
Ich wünsche mir weitere Audioinhalte, auch wenn nur Folieninhalte vorgelesen werden.	20,23%
Hyperlinks innerhalb einer Präsentation sind hilfreich.	70,37%
Hyperlinks zwischen verschiedenen Präsentation sind hilfreich.	57,41%

**Tab. 2.:** Umfrageergebnisse zu HTML-Präsentationen (n = 54 Teilnehmende).

Wie in Abschnitt «Ausgangssituation» angesprochen, fiel die Wahl des Autors auf HTML-Präsentationen mit Audioinhalten (anstelle von Videos) einerseits aufgrund ihrer Vorteile hinsichtlich ALMS-Framework und Single Sourcing, andererseits aber auch aufgrund subjektiver Vorlieben. Diesen subjektiven Vorlieben stellen die Zeilen 2 und 3 in Tabelle 2 die Präferenzen der Studierenden gegenüber. Mit 42,59% teilt die grösste Gruppe die Vorlieben des Autors, knapp vor 37,04%, die Videos bevorzugen würden, während sich die verbleibenden ca. 20% indifferent zeigen. Obwohl dieses Ergebnis nicht besonders deutlich ausfällt, bestärkt es den Autor in seinem gewählten Vorgehen.

Weitere Zeilen von Tabelle 2 zeigen, dass sich zahlreiche Studierende weitergehende Audio-Erläuterungen wünschten, was teilweise bereits während des Semesters aufgegriffen wurde, teilweise im kommenden Durchgang adressiert wird. Schliesslich äusserten sich die Studierenden positiv zu Hyperlinks innerhalb von sowie zwischen Präsentationen, wobei spekuliert werden kann, dass die positive Einschätzung zu Hyperlinks zwischen Präsentationen gegen Semesterende noch deutlicher ausgefallen wäre, da insbesondere Vorwärtsverweise (also Querbezüge zu noch nicht behandelten Themen) bei der initialen Erarbeitung eher hinderlich sind, im Zuge der Prüfungsvorbereitung aber relevant werden.

### Zusammenfassung und Ausblick

Die Ergebnisse dieses Beitrags können als Schritte zur Umsetzung von OER fördernden Aktionen des *Ljubljana OER Action Plan 2017* aufgefasst werden: Berichte von erfolgreichen OER-Einführungen helfen zunächst, ein Bewusstsein für OER als Alternative zu urheberrechtlichen verschlossenen Lehr- und Lernmaterialien zu schaffen, wobei insbesondere die Vermeidung redundanter Arbeit im Zuge der Erstellung qualitativ hochwertiger Materialien direkte Vorteile auf Seiten der Lehrenden verspricht. Für die Erstellung von OER stehen allerdings vielfältige Techniken und Werkzeuge zur Verfügung, deren Auswahl das Potenzial zur Weiternutzung und -entwicklung direkt beeinflusst. Aufbauend auf dem ALMS-Framework, gängigen Techniken der Software-Entwicklung und dem Single-Sourcing-Prinzip wurde in Abschnitt «Ausgangssituation» ein Anforderungskatalog entwickelt, der OER charakterisiert, die mit freier Software erstellt und organisationsübergreifend weiterentwickelt sowie plattformunabhängig genutzt werden können.

Diese Anforderungen werden im Kontext der gemeinschaftlichen Lehrbuchentwicklung unter Nutzung von LaTeX und Git in natürlicher Weise und ohne grossen Aufwand erfüllt, wie in Abschnitt «Entwickelte Inhalte und Software» dargestellt wurde. Für die Erstellung von zum Selbststudium geeigneten Präsentationen wurde dagegen mit *emacs-reveal* eine neue Software mit unterstützender Infrastruktur auf GitLab vorgestellt, die diese Anforderungen erfüllt. In Abschnitt «Erste Umfrageergebnisse» präsentierte Umfrageergebnisse geben Anlass, den präsentierten Ansatz weiterzuentwickeln.

Ein konkreter Aspekt für zukünftige Arbeiten besteht in der automatisierten Erzeugung der erläuternden Audiokommentare durch Text-To-Speech- (TTS-) Techniken (für eine Übersicht vgl. Siddhi, Verghese, und Bhavik 2017) aus Textvorlagen. Im dargestellten Projekt wurden Audioaufnahmen gezielt eingesprochen, was bei Korrekturen oder Ergänzungen im Semesterverlauf relativ aufwändige Audioschnittmassnahmen nach sich zog. Für zukünftige Iterationen der Vorlesung steht eine Evaluation freier TTS-Implementierungen hinsichtlich ihrer erzielten Sprachqualität

aus. Bei akzeptabler Qualität könnten die Audiokommentare automatisiert in der per Docker bereitgestellten GitLab-Infrastruktur aus Textdateien erzeugt werden, was jeglichen Aufwand für Audioaufnahme und -schnitt eliminieren würde.

Abschliessend sei angemerkt, dass dieser Artikel *eine* mögliche Philosophie von OER-Erstellung und -Weiterentwicklung im Single-Sourcing-Prinzip vertritt, und zwar eine, in der OER durch einfache Textdateien beschrieben werden, die unter Trennung von Layout und Inhalt offline bearbeitet werden können. Eine andere Philosophie wird von Auer, Khalili, und Tarasowa (2013) im Projekt SlideWiki.org vertreten, wo Präsentationen gemeinschaftlich im Wiki-Stil an einer zentralen Stelle unter Nutzung eines Online-WYSIWYG-Editors entwickelt werden. Angesichts des bestehenden Spektrums an Möglichkeiten zur OER-Gestaltung besteht Anlass zur Hoffnung, dass Lehrende mit unterschiedlichen Vorlieben «ihre» Lösung finden und einsetzen werden.

### Danksagung

Der Autor dankt dem Ministerium für Innovation, Wissenschaft und Forschung des Landes Nordrhein-Westfalen und dem Stifterverband für die Förderung im Rahmen eines Fellowships für Innovationen in der digitalen Hochschullehre, der Dekanin der Wirtschaftswissenschaftlichen Fakultät der WWU Münster, Frau Prof. Dr. Theresia Theurl, für die im Zusammenhang mit dem Fellowship gewährte Reduktion seines Lehrdeputats, ohne die das dargestellte Projekt nicht umsetzbar gewesen wäre, und Herrn Nicolas Pflanzl sowie einem anonymen Gutachter für kritische Anmerkungen zu einer Vorversion dieses Artikels, die zu wesentlichen Verbesserungen der Darstellung geführt haben.

### Literatur

- Andrade, António, et al. 2011. «Beyond OER: Shifting focus to open educational practices». Open Educational Quality Initiative (OPAL). <https://oerknowledgecloud.org/sites/oerknowledgecloud.org/files/OPAL2011.pdf>.
- Atkins, Daniel E., John Seely Brown, und Allen L. Hammond. 2007. «A Review of the Open Educational Resources (OER) Movement: Achievements, Challenges, and New Opportunities». Report to The William; Flora Hewlett Foundation. [http://www.oerders.org/wp\\_content/uploads/2007/03/a\\_review\\_of\\_the\\_open\\_educational\\_resources\\_oer\\_movement\\_final.pdf](http://www.oerders.org/wp_content/uploads/2007/03/a_review_of_the_open_educational_resources_oer_movement_final.pdf).
- Auer, Sören, Ali Khalili, und Darya Tarasowa. 2013. «Crowd-sourced Open Courseware Authoring with SlideWiki.org». *International Journal of Emerging Technologies in Learning (IJET)* 8 (1): 62–63. <https://doi.org/10.3991/ijet.v8i1.2539>.

- Bishop, Jacob Lowell, und Matthew A. Verleger. 2013. «The flipped classroom: A survey of the research». In *2013 ASEE Annual Conference & Exposition*, 1–18. <https://peer.asee.org/22585>.
- Chacon, Scott, und Ben Straub. 2014. *Pro Git*. 2. Apress. <https://doi.org/10.1007/978-1-4842-0076-6>.
- Ehlers, Ulf-Daniel. 2011. «Extending the territory: From open educational resources to open educational practices». *Journal of Open Flexible and Distance Learning* 15 (2): 1–10. <http://www.jofdl.nz/index.php/JOFDL/article/view/64>.
- Feller, Joseph, Brian Fitzgerald, Scott A. Hissam, und Karim R. Lakhani, Hrsg. 2005. *Perspectives on Free and Open Source Software*. MIT Press.
- Glushko, Robert J. 2015. «Collaborative Authoring, Evolution, and Personalization for a Transdisciplinary Textbook». In *Companion to the Proceedings of the 11<sup>th</sup> International Symposium on Open Collaboration*, 10:1–10:10. <https://doi.org/10.1145/2789853.2789867>.
- Hailperin, Max. 2016. *Operating Systems and Middleware – Supporting Controlled Interaction*. Revised edition 1.2.1.
- Hailperin, Max. 2017. *Operating Systems and Middleware – Supporting Controlled Interaction*. Revised edition 1.3. <https://gustavus.edu/mcs/max/os-book/>.
- Hefferon, Jim, und Albert Schueller. 2016. «Writing an Open Text». *The Mathematical Intelligencer* 38 (2). Springer: 6–9. <https://doi.org/10.1007/s00283-015-9606-1>.
- Hilton, John III, David Wiley, Jared Stein, und Aaron Johnson. 2010. «The four ‘R’s of openness and ALMS analysis: frameworks for open educational resources». *Open Learning* 25 (1). Taylor & Francis: 37–44. <https://doi.org/10.1080/02680510903482132>.
- Jung, Insung, Teruyoshi Sasaki, und Colin Latchem. 2016. «A framework for assessing fitness for purpose in open educational resources». *International Journal of Educational Technology in Higher Education* 13 (1). <https://doi.org/10.1186/s41239-016-0002-5>.
- Kerres, Michael, und Richard Heinen. 2014. «Open Educational Resources und schulisches Lernen: Das Zusammenwirken von Plattformen für Lernressourcen in informationell offenen Ökosystemen». In *MEDIEN – WISSEN – BILDUNG: Freie Bildungsmedien und Digitale Archive*, Petra Missomelius, Wolfgang Sützl, Theo Hug, Petra Grell, und Rudolf Kammerl (Hrsg.), 189–210. innsbruck university press. [https://www.uibk.ac.at/iup/buch\\_pdfs/freiebildungsmedien\\_web.pdf](https://www.uibk.ac.at/iup/buch_pdfs/freiebildungsmedien_web.pdf).
- McGee, Monnie, Lynne Stokes, und Pavel Nadolsky. 2016. «Just-in-Time Teaching in Statistics Classrooms». *Journal of Statistics Education* 24 (1): 16–26. <https://doi.org/10.1080/10691898.2016.1158023>.
- Merkel, Dirk. 2014. «Docker: Lightweight Linux Containers for Consistent Development and Deployment». *Linux Journal*. 2014 (239). Houston, TX: Belltown Media.
- Mruck, Katja, Günter Mey, Sandra Schön, Heiko Idensen, und Peter Purgathofer. 2013. «Offene Lehr- und Forschungsressourcen. Open Access und Open Educational Resources». In *L3T. Lehrbuch für Lernen und Lehren mit Technologien*, herausgegeben von Martin Ebner und Sandra Schön, 2. Auflage. epubli. <https://l3t.tugraz.at/index.php/LehrbuchEbner10/article/view/112>.



- Novak, Gregor M., Evelyn T. Patterson, Andrew Gavrin, und Wolfgang Christian. 1999. *Just-in-Time Teaching: Blending Active Learning and Web Technology*. Prentice Hall.
- Rockley, Ann. 2001. «The Impact of Single Sourcing and Technology». *Technical Communication* 48 (2). Society for Technical Communication: 189–93. [http://rockley.com/articles/Single\\_Sourcing\\_and\\_Technology.pdf](http://rockley.com/articles/Single_Sourcing_and_Technology.pdf).
- Schulte, Eric, und Dan Davison. 2011. «Active Documents with Org-Mode». *Computing in Science & Engineering* 13 (3): 66–73. <https://doi.org/10.1109/MCSE.2011.41>.
- Shaffer, Clifford A., Thomas L. Naps, und Eric Fouh. 2011. «Truly Interactive Textbooks for Computer Science Education». In *Proceedings of the Sixth Program Visualization Workshop*, 97–103. <https://people.cs.vt.edu/shaffer/Papers/pvw1.pdf>.
- Siddhi, Desai, Jashin M. Verghese, und Desai Bhavik. 2017. «Survey on Various Methods of Text to Speech Synthesis». *International Journal of Computer Applications* 165 (6). <https://doi.org/10.5120/ijca2017913891>.
- Stallman, Richard M. 1986. «What Is the Free Software Foundation?». *GNU's Bulletin* 1 (1). <https://www.gnu.org/bulletins/bull11.html#SEC4>.
- Thorogood, Miles. 2016. «slideDeck.js: A Platform for Generating Accessible and Interactive Web-Based Course Content». In *Proceedings of the 21<sup>st</sup> Western Canadian Conference on Computing Education*, 13:1–:5. WCCCE'16. New York, NY, USA: ACM. <https://doi.org/10.1145/2910925.2910941>.
- Vukovic, Boris, und Claude Martin. 2009. «Free and open source software (FOSS) and OER: Part I – An introductory note». In *Open Educational Resources: Conversations in Cyberspace*. UNESCO Publishing, 105–124. <https://unesdoc.unesco.org/ark:/48223/pf0000181682>.
- Wiley, David. 2014. «The Access Compromise and the 5th R». <https://opencontent.org/blog/archives/3221>.
- Zentrum für Hochschuldidaktik (DiZ), ed. 2016. «DiNa 12/2016». <https://diz-bayern.de/publikationen/dina/360-2016-dezember-hd-mint-sonderausgabe-wege-zum-verstaendnis-bauen>.