



Autor

Ron Lucke

### Vorwort

In Stud.IP Inhalte zu erstellen und durch multimediale Elemente zu bereichern begeistert immer mehr Stud.IP NutzerInnen. Seit mehr als vier Jahren ist dies mit Courseware möglich und wird an Hochschulen, beim Deutschen Fußballbund und in Volkshochschulen genutzt. In dieser Zeit hat sich das Plug-in durch den Input der NutzerInnen stark weiterentwickelt. Für viele ist Courseware heute ein fester Bestandteil der Arbeit mit Stud.IP.

Der Wunsch in Stud.IP Lerninhalte zu erstellen führte in der Vergangenheit zu zahlreichen Diskussionen und Workshops. Das Courseware Plug-in in den Kern von Stud.IP zu integrieren wurde aufgrund der verwendeten Technologien kontrovers diskutiert. Dieses Konzept beschreibt die Anforderungen an eine inhaltsbasierte Ansicht für Stud.IP. Als Grundlage dient Courseware, wobei die Weiterentwicklung von Courseware und technische Neuerungen miteinbezogen werden.



## **Funktionen**

Die inhaltsbasierte Ansicht soll Lehrenden und Studierenden die Möglichkeit geben Lerninhalte in Stud.IP zu erstellen, Lernfortschritte zu veranschaulichen und gemeinsam Ziele zu verfolgen. Abweichend von der jetzigen Struktur in Courseware soll hierarchische Struktur beliebig tief verschachtelbar sein. Lerninhalte sollen durch multimediale interaktive und kooperative Elementen sowie durch Aufgaben und externen Inhalten bereichert werden können. Neben der hierarchischen Struktur sorgen auf jeder Seite sogenannte Container für ein einheitliches Layout, welches sich in das Stud.IP Design nahtlos einfügt. Diese Container erlauben die Anordnung der Inhaltselemente auf moderne und leicht zugängliche Weise.

Elemente lassen sich dort bearbeiten, löschen und anordnen wo sie erscheinen, so bleibt der inhaltliche Zusammenhang erhalten. Inhalte können mehrsprachig erfasst werden. Das geeignete Inhaltselement kann mit Hilfe eines Assistenten gefunden werden. Um den Workflow noch weiter zu unterstützen, können aus den diversen Inhaltselementen Favoriten gewählt werden. Für umfangreichere Sortier- und Kopiervorgänge gibt es eine spezielle Ansicht der Struktur und ihrer Inhalte. Strukturen und Inhalte lassen sich sichern und in anderen Veranstaltungen wiederverwenden, somit ist die Nachhaltigkeit von Lehrmaterial gewährleistet.

Lernen und Lehren kann unterschiedliche Formen annehmen, die inhaltsbasierte Ansicht soll Lehrende und Studierende hierbei unterstützen. Ob Unterstützung von Präsenzlehre, Blended Learning, E-Portfolio, kollaboratives Lernen oder Webinare, durch Optionen wie Sequentiellerablauf, Schreib- und Leserechte für Gruppen und TeilnehmerInnen, Scrollytelling, Diskussionsübersicht etc. sollen eine Vielzahl von Szenarien abgebildet werden können. Neue Trends und Technologien können durch Plug-ins schnell integriert werden, diese können Inhaltselemente und Container für besondere Layouts liefern.

Die Gestaltung des Layouts sowie zahlreiche Funktionen unterstützen die Lehrenden dabei die Übersicht zu behalten. Lernende können in der Übersicht einsehen welche Inhalte vorhanden sind und welche davon schon bearbeitet wurden. Als Einstiegspunkt dient immer das zuletzt gewählte Strukturelement, somit kann direkt weiter gearbeitet werden. Eine Suche in der Sidebar sowie eine Übersicht über neue Inhalte steigern außerdem die Möglichkeit Lerninhalte schnell zu finden. Um die Übersicht zu behalten an welcher Stelle im Lerninhalt man sich befindet wird im oberem Bereich eine Breadcrumb Navigation angezeigt. Außerdem befindet sich in der Sidebar eine kompakte Darstellung der Elemente im Umfeld, diese Ansicht passt sich dynamisch an. Beide Elemente sind dezent eingebunden und lenken nicht vom eigentlichen Inhalt ab. Für noch Fokus auf den Inhalt sorgt die Vollbildansicht.

Die inhaltsbasierte Ansicht beschränkt sich nicht nur auf Veranstaltungen und Studiengruppen. NutzerInnen sollen im Bereich "eigene Inhalte" die Möglichkeit bekommen den vollen Funktionsumfang des Erstellens von Inhalten in Stud.IP zu nutzen.

Motivierend und unterstützend wirkt der Companion, ein persönlicher Assistent. Er zeigt das erreichen von vorgegebenen Zielen in Form von Badges, informiert über den Lernfortschritt und hilft beim erstellen und erkunden der Inhalte.

## Aufbau

Strukturelemente bilden die Baumstruktur und können beliebig tief verschachtelt werden, also beliebig viele Strukturelement enthalten. Strukturelemente können außerdem beliebig viele Container enthalten. Container wiederum enthalten beliebig viele Inhaltselemente (Blöcke). Container können keine Container und Blöcke können keine Blöcke enthalten.

#### Strukturelemente

Ein Strukturelement hat einen Kurz- und einen Langtitel. Es kennt immer das übergeordnete Strukturelement und seine Position.

Außerdem enthält es Informationen über Sichtbarkeit, Lese- und Schreibrechte von Gruppen sowie externe Relationen. Eine externe Relation ist zum Beispiel die Verknüpfung mit einem Thema der Veranstaltung. Das Strukturelement verfügt über Metadaten, diese können zur Beschreibung des Inhalts genutzt werden. Sowohl die Metadaten als auch die Titel sind lokalisierbar. Für unterschiedliche Layout Optionen kann dem Strukturelement ein Bild zugeordnet werden.

#### Container

Ein Container kennt immer das übergeordnete Strukturelement und seine Position. Außerdem kennt es den Bereich auf der Seite in dem er erscheinen soll. Anhand seines Klassennamens und des Typen lässt sich erkennen welche Art von Layout er mitbringt.

Bereiche sind derzeit die Sidebar und der zentrale Inhaltsbereich.

#### Blöcke

Blöcke sind die Inhaltselemente und sorgen für die Darstellung des Inhalts. Jeder Block ist für die Aufnahme des Inhaltes, spezielle Einstellungen und die Darstellung verantwortlich. Hierbei ist ein einheitliches Layout und eine konsistente Benutzeroberfläche zu implementieren. Ein Block kennt den übergeordneten Container sowie seine Position darin. Optional kann ein Block von Lehrenden und NutzerInnen kommentiert werden.

Neben eines grundlegenden Standard-Blöcken sind mindestens drei besondere Block-Typen zu identifizieren, Multimedia-Blöcke, Kooperative-Blöcke und Aufgaben-Blöcke. Besondere Funktionen wie Zugriff auf den Dateibereich oder Auswahl von Gruppen werden entsprechend zur Verfügung gestellt. Somit muss nicht jeder Block eine eigene Lösung implementieren.

Nutzerfortschritt und Nutzer bezogene Daten werden für die Blöcke gesondert behandelt und lassen sich über den Block und Nutzer eindeutig identifizieren.

#### Element hinzufügen

Container und Blöcke werden in einer Auswahl bereitgestellt, jedes Element besitzt ein Icon und einen Beschreibungstext, über einen Link können mehr Informationen zum Element abgerufen werden (Nutzung, Einsatzszenarien, Beispiele, Videos). Der Companion (siehe unten) unterstützt bei der Auswahl der Elementes. Außerdem können Elemente als Favorit markiert werden und somit in einer persönlichen Auswahlliste angezeigt werden.

### Inhaltselemente

Wie schon zuvor beschrieben lassen sich Blöcke aufgrund ihrer Anforderungen in verschiedene Gruppen unterteilen. Wobei die Spezialisierung durch Vererbung erfolgt. Ein Standard-Block verfügt über Inhalt, Einstellungen und Layout. In Courseware wären es derzeit folgende Blöcke: Bestätigung, Diagramm, Embed, externer Inhalt, Forum, Freitext, Link, Merksatz, OpenCast, Quellcode, Schreibmaschine und Termin.

Multimedia-Blöcke nutzen Dateien aus dem Dateibereich oder über eine URL. Funktionen zur Bereitstellung, Zugriffsprüfung, Verfügbarkeit etc. bringt diese Art von Block mit. In Courseware wären es derzeit folgende Blöcke: Audio, Audio Galerie, Bildvergleich, Download, Galerie, interaktives Video, Leinwand, Lernkarten, PDF mit Vorschau, Video.

Als kooperative Blöcke lassen sich auf bestimmte Gruppen begrenzen und haben hierfür bestimmte Funktionen. Derzeit Verfügt Courseware über die Blöcke Kommentare & Diskussion und Audio Galerie welche sich als kooperativ bezeichnen lassen könnten. Als Weiterentwicklung sind eine Video Galerie und ein Etherpad Block angedacht. Außerdem könnte man eine Erweiterung des Leinwand-Blocks in Betracht ziehen.

Für Aufgaben arbeitet Courseware im Moment nur mit dem Vips Plug-in zusammen. Aufgaben-Blöcke sollen als Grundgerüst dienen um anderen Aufgaben-Tools das bereitstellen eines Blocks zu ermöglichen.

Das Layout der Inhaltselemente soll einheitlich sein, dies gilt besonders für Blöcke die durch Plug-ins eingebunden werden. Jedes Inhaltselement verfügt über ein Layout zum bearbeiten, eins für die Anzeige im zentralen Inhaltsbereich und eins für die Anzeige in der Sidebar. Eine korrekte Anzeige in verschiedenen Container-Layouts muss dabei stets möglich sein.

Den aktuellen Funktionsumfang von Courseware Blöcken können Sie <u>hier</u> nachlesen. Eine Liste der bestehenden und geplanten Inhaltselemente befindet sich im Anhang.

#### Sichtbarkeit

Strukturelemente haben ein Veröffentlichungs- und ein Widerrufsdatum welches die Sichtbarkeit steuert. Alle Elemente unterhalb des Strukturelements sind sichtbar bzw. unsichtbar. Dies gilt auch für die Einschränkungen von Gruppen. Ist ein Strukturelement nur für eine bestimmte Gruppe Lesbar so ist es für andere Teilnehmer nicht sichtbar.

#### Schreibrechte

Strukturelemente können für TeilnehmerInnen und Gruppen freigeben werden, sodass sie Elemente hinzufügen und bearbeiten können. Diese Strukturelement sind in der Ansicht für TeilnehmerInnen besonders gekennzeichnet und in der Übersicht hervorgehoben.

## **Container-Layouts**

#### Liste

Stellt die Inhaltselemente geordnet untereinander da. Dieses Layout kann in der Sidebar oder im zentralen Inhaltsbereich genutzt werden.

#### **Tabs**

Angelehnt an die JQuery-UI Tabs können mehrere Blöcke unter einem Navigationsreiter zusammengefasst werden. Dieses Layout kann in der Sidebar oder im zentralen Inhaltsbereich genutzt werden.

#### Accordion

Angelehnt an die JQuery-Ul Accordion können mehrere Blöcke unter einem ausblendbaren Bereich zusammengefasst werden. Dieses Layout kann in der Sidebar oder im zentralen Inhaltsbereich genutzt werden.

#### Scrollytelling

In diesem Speziellen Layout wird die Sidebar ausgeblendet und kann bei bedarf angezeigt werden. Die Inhaltselemente werden zeilenweise dargestellt. Eine Zeile kann ein zentriertes Inhaltselement in definierten Breiten (klein, mittel, groß, volle Breite) enthalten. Außerdem kann am Zeilenanfang und -ende jeweils ein Block mit definierter Größe gesetzt werden.

In einer Vollbildansicht wird die komplette Stud.IP Navigation ausgeblendet und nur der zentrale Inhaltsbereich angezeigt.

#### **Branching Scenario**

In diesem besonderen Layout werden Blöcke abhängig von Entscheidungen der Nutzerlnnen angezeigt. Informationen zu der Gabelung trägt der Container. Es soll möglich sein eine weit verzweigte Baumstruktur zu erzeugen. Der Lehrende erhält hierfür eine Oberfläche in der per Drag-And-Drop Elemente angeordnet werden können. Dieses Layout kann nur im zentralen Inhaltsbereich verwendet werden.

#### Weitere Layouts

Neue Layout-Typen können per Plug-in eingebunden werden. Somit kann auf standortspezifische Anforderungen reagiert werden.

### Blöcke

Bereits bestehende und geplante Blöcke sind im Anhang aufgelistet.

Die Basis-Blockklasse stellt das einheitliche Design und den grundlegende Funktionsumfang bereit. Von dieser Klasse können sich spezielle Blockklassen ableiten, dass Design und der Workflow muss hierbei konsistent bleiben. Ein wichtiger weiterer Blockklassentyp sind Blöcke die Dateien verwenden, entweder aus dem Dateibereich oder von anderen Quellen. Diese Blöcke verfügen über einen kleinen Dateibrowser, in dem eine Datei aus dem Dateibereich ausgewählt werden kann. Außerdem kann ein solcher Block per Drag-And-Drop mit einer Datei bestückt werden. Es sind alle nötigen Funktionen für Datenspeicherung, Export, Import und Dateibereitstellung vorhanden. Blöcke die per Plug-in nachgerüstet werden haben eine eigene Klasse.

Inhalte eines Blocks lassen sich mehrsprachig erfassen. Multimedia Inhalte lassen sich ebenfalls mehrsprachig hinterlegen.

Alle Blöcke haben eine Liste von Kategorien denen sie angehören. Diese Kategorien helfen beim auswählen eines Blockes.

Neben anpassbaren Blöcken soll es in Zukunft auch statische Blöcke geben. Diese stellen Informationen aus dem System bereit. Ein Administrator soll die Möglichkeit haben Blöcke mit statischem HTML Inhalt zu erstellen, diese haben einen Titel, eine Beschreibung und ein Icon. Somit muss für relativ simple Blöcke kein Plug-in geschrieben werden.

#### Layout

Jeder Block verfügt über einen Inhaltsbereich und ausklappbare Features. Ausgeklappt werden können die Bearbeitungsansicht, ein Kommentarfeld (optional für Lehrende und/oder TeilnehmerInnen) und eine Blockinformation die Metadaten und Hilfe enthält. Im gleichen Design wie die Knöpfe zum ausklappen sind Features wie Lesezeichen, kopieren in die Zwischenablage, sichtbar schalten und die "direkter Export" Funktion.

#### Sichtbarkeit

Jeder Block kann einzeln von Nutzerlnnen mit Schreibrechten unsichtbar geschaltet werden.

#### Blöcke hinzufügen

Beim Hinzufügen erhalten NutzerInnen eine Übersicht der Blöcke die sich nach Kategorien filtern lässt. Außerdem können Block Favoriten gesetzt werden und somit leichter ausgewählt werden. Bei der Blockauswahl kann falls gewünscht der Companion (siehe unten) Unterstützung leisten. Er führt durch gezielte Fragen nach dem gewünschten Einsatz zu einer begrenzten Anzahl von sinnvollen Blöcken.

## Sichern, Wiederverwenden und Verteilen — OER

Erstellte Inhalte zu sichern, wiederherstellen und zu verteilen erfordert in technischer Hinsicht einen Import- und Exportmechanismus, sowie die Gewährleistung der Kompatibilität.

Zum sichern oder weitergeben der Inhalte wird ein einheitliches Exportformat verwendet. In einer Datendatei wird die Struktur, die verwendeten Elemente und der Inhalt der Blöcke gespeichert. Zum Inhalt gehörende Dateien werden gesammelt und mit der Datendatei in einer Archivdatei komprimiert gespeichert. NutzerInnen können entscheiden ob die gesamten Inhalte oder nur bestimmte Teile exportiert werden sollen. In einer speziellen Oberfläche können NutzerInnen per Drag-And-Drop die gewünschten Inhalte auswählen.

Ein solches Archiv kann wieder importiert werden, somit können erstellte Inhalte beliebig oft wiederverwendet werden. Liegen die Inhalte noch im System muss kein Umweg über den Export und Import gegangen werden. Inhalte aus eigenen Veranstaltungen können direkt in die ausgewählte Veranstaltung kopiert werden. Hierbei können Nutzerlnnen wieder per Drag-And-Drop auswählen und die Inhalte an die gewünschte Stelle einfügen.

Wird der Inhalt einer Veranstaltung als "freigegeben" markiert kann dieser auch von anderen NutzerInnen kopiert werden. Somit lassen sich Inhalte innerhalb einer Stud.IP Installation verteilen. Freigegeben werden kann der Inhalt für das gesamte System, Fachbereiche oder einzelnen NutzerInnen. Soll das Verteilen über einen Standort hinausgehen so lässt sich der Inhalt für den Marktplatz freigeben. Inhalte mit dieser Markierung werden auf einem gesonderten Marktplatz hochgeladen und bereitgestellt. Inhalte des Marktplatzes können dort direkt eingesehen, als Archiv heruntergeladen und direkt in Stud.IP in die ausgewählte Veranstaltung kopiert werden.

Diese Funktionalitäten gelten für "eigene Inhalte" analog. NutzerInnen erhalten somit die Möglichkeit Inhalten zu sammeln oder als Vorlage für eigene Inhalte zu verwenden.

Die Stabilität dieser Mechanismen muss gewährleiste sein. Container oder Blöcke aus Plug-in sollten wenn möglich importiert und exportiert werden. Ist dies nicht möglich wird ein Platzhalter gesetzt und ein Hinweis angezeigt. Der Import von Courseware Archiven muss ebenfalls gewährleistet sein. Die alte Struktur muss auslesbar sein und übersetzt werden.

Ein spannendes Feature ist der Import und Export von Formaten anderer Plattformen. Der Import eines Moodle oder SCORM Moduls würde es Dozenten erleichtern ihre Lerninhalte mitzubringen. Der Export in HTML5 und PDF ermöglicht es die Lerninhalte an einem beliebigen Ort einzusetzen.

Ein Export kann auch direkt am Block, Container oder Strukturelement vorgenommen werden. Hierfür sind an entsprechenden Stellen Knöpfe angebracht. Der Export kann dann wahlweise als Courseware Archiv, HTML5 oder PDF erfolgen.

# Metadaten

Bei jedem Export in ein Courseware Archiv werden Metadaten mitgeliefert.

Courseware	Syllabus → i18n		
	Autor (wird aus System übernommen)		
	Erstellungsdatum (wird vom System gesetzt)		
	Lizenztyp		
	Thema → i18n		
	Voraussetzungen → i18n		
	Einsatzszenario		
Strukturelement	Autor (wird aus System übernommen)		
	Erstellungsdatum (wird vom System gesetzt)		
	Thema → i18n		
	Nutzergruppe → i18n		
	Art der Inhalte (Medientypen)		
	Lizenztyp		
	geschätzter zeitlicher Aufwand		
	Lehr-/Lerntyp		
	Voraussetzungen → i18n		
Blöcke	Autor (wird aus System übernommen)		
	Erstellungsdatum (wird vom System gesetzt)		
	Medientyp (wird vom System gesetzt)		
	Beschreibungen → i18n		
	Lizenztyp		
	geschätzter zeitlicher Aufwand		

## **Gamification**

## **Call to Action & Showstopper**

Um den Inhalt aufzulockern und die TeilnehmerInnen zu motivieren können an bestimmten Punkten oder bei bestimmten Events Showstopper eingeblendet werden. Als Vollbildeinblendung kann ein plakativer Text, ein animierter Text, ein Bild oder ein Video angezeigt werden. Zeitpunkt und Inhalt sind frei wählbar. Um die Motivation noch weiter zu steigern kann sich der Companion (siehe unten) mit einer Botschaft (Text, Bild, Video etc.) melden. Dieser kann auch eventbasiert den Fortschritt animiert darstellen.

### XP-Punkte

NutzerInnen können systemweit XP-Punkte sammeln. Punkte werden für das erreichen von Zielen und das meistern von Herausforderungen vergeben. XP-Punkte können zum freischalten bestimmter Funktionen verwendet werden. Diese Funktionen haben aber nur spielerischen Charakter und verändern die Nutzung nur rein optisch.

## Ziele, Trophäen & Badges

Lehrende können Ziele festlegen, z.B. das betrachten eines Videos oder das verfassen eines Kommentars. Ein Ziel verfügt über ein Badge, XP-Punkte, eine Beschreibung und eine Trophäen-Stufe. Es gibt fünf Stufen, gewöhnlich, ungewöhnlich, selten, episch und legendär. Badges können vom Lehrenden aus einer Vielzahl von Vorlagen ausgewählt werden, der Upload eines Bildes ist auch möglich. Die Beschreibung des Zieles kann mehrsprachig hinterlegt werden. Außerdem kann ein Ziel versteckt sein, es wird erst sichtbar wenn es erreicht wurde. Erreichte und verfügbare Ziele können TeilnehmerInnen in einer Übersicht einsehen. Beim erreichen eines Ziels werden NutzerInnen vom Companion (siehe unten) darüber informiert.

### Herausforderungen

Anders als Ziele sind Herausforderungen in einem bestimmten Zeitraum zu erfüllen. Somit lassen sich z.B. wöchentliche Aufgaben erstellen. Eine Aufgabe kann das erstellen eines Elements, die kollaborative Nutzung oder das bestehen eines Blocks sein. Eine Aufgabe kann auch aus mehrerer Aufgabenteile bestehen. Für das Meistern einer Herausforderung erhalten Nutzerlnnen XP-Punkte.

## Companion

Der Companion ist eine virtuelle Unterstützung für Lehrende und Lernende. Die Erscheinung des Companion kann aus einer Anzahl von Vorgaben ausgewählt werden. Mit dem Einsatz von XP-Punkten kann das Aussehen verändert werden.

NutzerInnen zeigt sich der Companion entweder animiert oder mit einem Avatar in einer Art Chat-Box. Falls in einer Veranstaltung ein Syllabus hinterlegt wurde wird dieser vom Companion genutzt um NutzerInnen zu Beginn des Kurses Informationen über das Thema, den Aufbau, die Ziele, die Dauer etc. zu liefern. Der Companion taucht an vielen Stellen auf an denen NutzerInnen Unterstützung benötigen könnten, auf der Übersichtsseite oder dem Dashboard für Lehrende, beim erstellen von Blöcken, in Showstoppern und bei erreichen von Zielen. Eine systemweite Abschaltung des Companion ist für jeden Nutzer und jede Nutzerin möglich.

### **Timeline**

Erreichte Ziele und Herausforderungen werden in der Timeline wie der Name schon sagt auf einem Zeitstrahl dargestellt. Dieser Zeitstrahl ist für jede Veranstaltung und unter meine Inhalte für das gesamte System verfügbar. Letztere Variante ist filterbar, sodass man z.B. nur ein bestimmtes Semester betrachten kann.

#### Quizduell

Um das Wissen zu einem Thema kompetitiv zu testen können TeilnehmerInnen die gleichzeitig online sind Fragen aus einem Pool beantworten. GewinnerInnen eines Duells erhälten XP-Punkte.

### **Smart Autoplay**

Zur Vereinfachung des Konsums von Inhalten kann das "smart Autoplay" aktiviert werden. Das nächste Strukturelement wird automatisch angezeigt sobald das aktuelle komplett betrachtet wurden. Kriterien sind hierbei unter anderem Audio oder Video wurde vollständig abgespielt, eine Galerie wurde komplett betrachtet, alle Aufgaben wurden abgegeben etc. .

Es wird am Ende der Seite ein Button mit der Aufschrift "weiter" und einem Timer angezeigt. Läuft der Timer ab wird auf die nächste Seite navigiert. Der Timer startet erst wenn alle Elemente auf der Seite "genutzt" wurden. NutzerInnen können "smart Autoplay" global abschalten.

## Übersichtsseite

Auf der Übersichtsseite erhalten TeilnehmerInnen Informationen über den Fortschritt, Ziele und Herausforderungen, den Companion, neue Inhalte und aktuelle Ereignisse. Um den Fortschritt zu veranschaulichen wird die oberste Ebene der Strukturelement in Form von Kacheln dargestellt, diese zeigen ein Bild, den Titel und den Fortschritt. Durch anklicken einer Kachel werden die darunterliegenden Strukturelement in gleicher Form dargestellt. Aktuelle Ereignisse werden in einem Feed angezeigt. Hier können Nutzerlnnen Antworten auf Kommentare, Kommentare zu erstellten Inhalten, Bewertungen von Übungen und neue Inhalte sehen. Zuletzt erreichte Ziele und Herausforderung werden in einer kleinen Box angezeigt, durch anklicken werden mehr Details angezeigt. Ziele und Herausforderungen werden nacheinander aufgelistet, zusätzlich lässt sich hier in die Timeline Ansicht wechseln.

In einer weiteren Box wird der Companion angezeigt, er bewegt sich in zufälligen Mustern und teilt Informationen mit. Durch anklicken des Companion gelangt man zu dessen Einstellungen. Hier kann das Aussehen angepasst werden.

## E-Portfolio & eigene Inhalte

Nutzerinnen und Nutzer erhalten mit der Funktion "eigene Inhalte" die Möglichkeit Inhalte in einem privaten Bereich zu erstellen und zu sammeln. Teile dieser Inhalte können in Veranstaltungen eingebunden oder zum kopieren freigeben werden. Eingebundene Inhalte können nur von der Besitzerin bzw. dem Besitzer verändert werden. Ebenso wie in Veranstaltungen können Inhalte als PDF oder HTML5 exportiert werden. Somit ist die Mitnahme und Wiederverwendbarkeit der Inhalte gewährleistet.

In Veranstaltungen können Strukturelemente für TeilnehmerInnen oder Gruppen zum schreiben und/oder lesen freigeben werden. Hierdurch ist es möglich gemeinsam Inhalte zu erstellen und zu bearbeiten.

### Zwischenablage

Elemente lassen sich zwischen Veranstaltung und eigenem Inhalt leicht über die Zwischenablage kopieren. Über einen Schalter am Element lässt sich dieses in die Zwischenablage legen. Die Zwischenablage wird in den eigenen Inhalten in einem gesonderten Bereich dargestellt, dort lassen sich die Elemente einsehen und aus der Zwischenablage entfernen. Beim Hinzufügen von Elementen werden Inhalte aus der Zwischenablage zum Einfügen angeboten. Es können nur eigene Elemente oder zum kopieren freigeben Elemente in die Zwischenablage gelegt werden.

## Lesezeichen

Struktur Elemente und Blöcke lassen sich mit Lesezeichen markieren. Lesezeichen können über einen Schalter am Element gesetzt und in einem extra Bereich der eigenen Inhalte eingesehen werden. In diesem Bereich können Blöcke betrachtet werden, Strukturelemente werden in einer Preview dargestellt. Für jedes Element wird ein Link in die Veranstaltung an die entsprechende Stelle generiert. Lesezeichen sind read-only, das bedeutet nur BesitzerInnen können die Elemente bearbeiten. Dargestellt wird immer der aktuelle Zustand des Elements und nicht der zum Zeitpunkt des Setzen des Lesezeichens. Lesezeichen lassen sich nach frei definierbaren Kategorien sortieren.

## **Courseware Analytics**

Um Optimierungspotential zu erkennen und TeilnehmerInnen besser zu unterstützen wird Lehrenden eine Vielzahl von Tools zur Verfügung gestellt.

Der bisherige Nutzerfortschritt wird durch Daten die von Blöcken und Strukturelementen erzeugt werden ausgebaut. Durch Auswertung bis hin zur Blockebene lassen sich Details wie durchschnittliche Abspielzeit von Videos und Audiodateien, setzen von Lesezeichen, Häufigkeit der Aufrufe von Strukturelementen und Blöcken und die durchschnittliche Betrachtungszeit einer Seite anzeigen. Außerdem lässt sich der Nutzerfortschritt für Blöcke feiner einstellen, optional lassen sich Zielwerte für Audio und Video Blöcke setzen, eine Aufnahme in der Audio Galerie ist nur bestanden wenn n Sekunden aufgenommen wurden, ein Download Block ist nur bestanden wenn auch der Download ausgelöst wurde und für Quiz Blöcke gibt es optional einen Schwellenwert bei dem der Block bestanden wird.

Weitere nützliche Informationen sind die Bewertungen von TeilnehmerInnen für Elemente, die Nutzung von Lernpfaden sowie die Veranschaulichung der täglichen Nutzung und der Absprungrate.

Zusammengefasst werden die Informationen auf dem Dashboard für Lehrende. Auf einer Übersicht werden Grunddaten wie Teilnehmerzahl, Teilnehmernutzung, Anzahl von Elementen etc. dargestellt. Außerdem wird ein Fortschrittsdiagramm und ein Activity Feed angezeigt. Letzterer informiert über letzte Änderungen (wo sind welche neuen Inhalte entstanden), neue Beiträge in Diskussionen und neue Bewertungen von TeilnehmerInnen. Auf separaten Seiten lassen sich die jeweiligen Detailansichten zu den Bereichen einsehen. Über das Dashboard gelangt man auch zu den grundlegenden Einstellungen für die Veranstaltung.

## Lernpfade

Lernpfade sind ein spannendes Thema dem sich Courseware nicht länger entziehen möchte. Lehrende können in einem separaten Bereich des Dashboards Lernpfade anlegen und organisieren. Das setzen von Metadaten an Blöcken und Strukturelementen vereinfacht die automatisch Erzeugung von Lernpfaden.

TeilnehmerInnen erhalten bei aktivierten Lernpfaden eine zusätzliche Navigation neben der Strukturübersicht. Diese Navigation fokussiert sich auf den gewählten Lernpfad. Außerdem macht der Companion am Ende eines Strukturelements Vorschläge für die weitere Navigation. Zu Beginn wird abgefragt welcher Lernpfad beschritten werden soll, diese Auswahl lässt sich auf der Übersichtsseite für TeilnehmerInnen jederzeit ändern.

## Anpassung aktueller Stud.IP Strukturen

Für Stud.IP Werkzeuge und Plug-ins kann ein Block bereitgestellt werden. Für Funktionen die nicht bereitgestellt werden kann über eine Plug-in-Schnittstelle neue Block-Typen hinzugefügt werden. Da manche Werkzeuge bzw. Plug-ins zu komplex sind ist eine Ansicht für den Lehrenden meist notwendig. Für Lernende sollten die Navigationspunkte zu diesen Elementen ausblendbar sein. Hier bietet sich die "mehr" Seite an. Dort soll die Sichtbarkeit für die Werkzeuge sowie deren Anzeigename einstellbar sein.

Blubber sollte einen eigenen Block mitbringen und zeigt Nutzerlnnen welche Blöcke diskutiert werden. Vergleichbar mit dem Block Kommentare & Diskussionen in der aktuellen Courserwareversion soll Blubber Nutzerlnnen die Möglichkeit geben sowohl direkt im Inhalt über den Block zu kommunizieren, als auch über eine Übersichtsseite. Diese Übersichtsseite zeigt die Stellen in denen Blubber-Blöcke verwendet werden und die Kommunikation die dort stattfindet. Gesprächsverläufe die seit dem letzten Login entstanden sind werden hervorgehoben.

Für das produktive Arbeiten werden zusätzliche Ordnertypen im Dateibereich benötigt. Um sicherzustellen das Dateien, die zu Inhalten gehören welche unter bestimmten Umständen erst eingesehen werden dürfen, nicht vorher eingesehen werden können wird ein unsichtbarer Ordner benötigt. Dieser Ordner kann nur vom Lehrenden gesehen und bearbeitet werden, die Inhalte können aber von den Lernenden eingesehen werden, wenn diese in einem Block verwendet werden oder der Link bekannt ist.

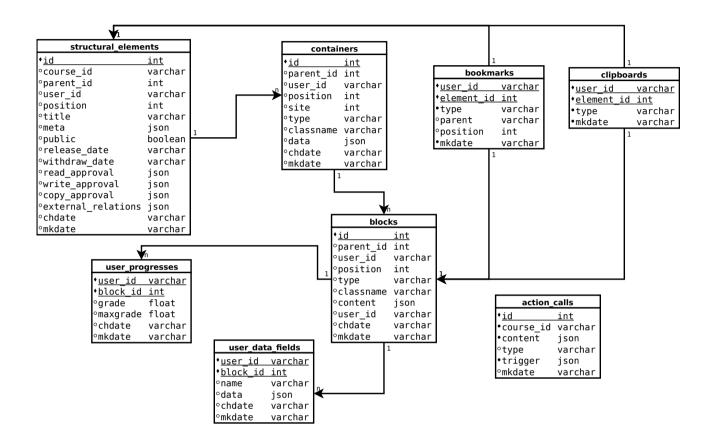
Außerdem soll es einen Ordner geben der direkt mit einem Block verbunden ist, der konditionale Ordner. Der Block des konditionalen Ordners ist nur dann sichtbar wenn der Ordner es im Dateibereich ist. Umgekehrt ist der Ordner auch nur dann sichtbar wenn der Block für den Lernenden in der Inhaltsansicht sichtbar ist.

#### Plug-ins

Container und Blöcke sollen per Plug-ins nachgerüstet werden können. Hierzu entsteht eine entsprechend Schnittstelle. Ein solches Plug-in muss die PluginBlock bzw. die PluginContainer Klasse implementieren. Ferner müssen Im- und Export definiert und die notwendigen Komponenten mitgeliefert werden.

# **Technische Beschreibung**

### Datenbankstruktur



strucutral_elements		
♦ id	int	primary key
course_id	varchar	ID der Veranstaltung zu dem das Strukturelement gehört
parent_id	int	ID des Strukturelements oberhalb dises Strukturelements
user_id	varchar	ID des Nutzers der diesen Block erstellt hat
position	int	Position in der Reihenfolge von benachbarten Strukturelementen
title	varchar	Lesbarer Name der in der Navigation erscheint
meta	json	Bereich für Daten wie Beschreibung, Informationen zum Autor etc.
public	boolean	Öffentlich Sichtbar oder nur für den Besitzer
release_date	varchar	Veröffentlichungsdatum
withdraw_date	varchar	Widerrufsdatum
read_approval	json	Liste von Gruppen und Teilnehmern die das Strukturelement sehen dürfen
write_approval	json	Liste von Gruppen und Teilnehmern die das Strukturelement bearbeiten dürfen
copy_approval	Json	Liste von Nutzern und Bereichen die das Strukturelement kopieren dürfen
external_relations	json	Verknüpfung mit anderen Stud.IP Elementen, z.B. Themen
chdate	varchar	Änderungsdatum
mkdate	varchar	Erstellungsdatum

containers		
<b>♦</b> id	int	primary key
parent_id	int	ID des Strukturelements oberhalb dises Strukturelements
user_id	varchar	ID des Nutzers der diesen Block erstellt hat
position	int	Position in der Reihenfolge von benachbarten Strukturelementen
site	int	Bereich der Seite in dem der Container erscheinen soll
type	varchar	Type des Containers
classname	varchar	PHP Klassenname
data	json	Weitere Daten zu diesem Container, z.B. Einstellungen
chdate	varchar	Änderungsdatum
mkdate	varchar	Erstellungsdatum

blocks		
<b>♦</b> id	int	primary key
parent_id	int	ID des Strukturelements oberhalb dises Strukturelements
user_id	varchar	ID des Nutzers der diesen Block erstellt hat
position	int	Position in der Reihenfolge von benachbarten Strukturelementen
type	varchar	Type des Blocks
classname	varchar	PHP Klasse des Blocks
content	json	Inhalt
user_id	varchar	User ID des Nutzers der diesen Block erstellt hat
chdate	varchar	Änderungsdatum
mkdate	varchar	Erstellungsdatum

user_data_filelds		
◆ user_id	varchar	primary key
♦ block_id	int	primary key
name	varchar	Bezeichnung der gespeicherten Daten
data	json	Nutzerspezifische Daten für einen Block
chdate	varchar	Änderungsdatum
mkdate	varchar	Erstellungsdatum

user_progresses		
◆ user_id	varchar	primary key
♦ block_id	int	primary key
grade	float	Note des Nutzers für diesen Block
max_grade	float	Maximale Note für diesen Block
chdate	varchar	Änderungsdatum
mkdate	varchar	Erstellungsdatum

bookmarks		
◆ user_id	varchar	primary key
◆ element_id	int	primary key
◆ type	varchar	Block oder Strukturelement
parent	varchar	Bezeichnung der Kategorie
position	int	Position innerhalb der
mkdate	varchar	Erstellungsdatum

clipboards		
◆ user_id	varchar	primary key
◆ element_id	int	primary key
◆ type	varchar	Block oder Strukturelement
mkdate	varchar	Erstellungsdatum

action_calls		
<b>♦</b> id	int	primary key
course_id	varchar	ID der Veranstaltung zu dem das Call-To-Action-Event gehört
content	json	Inhalt
type	varchar	Fullscreen, animated Companion or chat Companion
trigger	json	Event das Call-To-Action-Event auslöst
mkdate	varchar	Erstellungsdatum

### **SORM**

Für die in der Datenbankstruktur beschriebenen Tabellen gibt es jeweils ein Model. Die Models für strucutral\_element, container und block können sich am aktuellen Block-Model von Courseware orientieren, user\_data\_fields kann eine angepasste Version vom Field-Model werden und user\_progress kann übernommen werden.

Für strucutral\_elements sind Funktionen für die Sichtbarkeit und die Schreibrechte zu implementieren.

## **UI-Komponenten**

Zu den Elementen der inhaltsbasierten Ansicht sind Komponenten zu erstellen. Diese Komponenten bündeln die MVC-Verantwortlichkeiten in einem Paket.

## **Templates & Javascript**

Für den clientseitigen Aufbau wird vue.js verwendet. Dokumentstruktur und dynamisches Verhalten wird durch die vue Komponenten und Events gebildet.

### JSON-API & REST API

Für einen reibungslosen Ablauf in Kombination mit vue.js werden entsprechende Routen benötigt an die Daten gesandt und von denen Daten abgeholt werden können. Hierzu wird sich an der aktuellen Courseware-Version sowie kürzlich entwickelten Plug-ins die auf vue.js basieren orientiert.

## **Anhang**

### Liste der Inhaltselemente

Bestehende Courseware Blöcke

Audio spielt eine Audiodatei aus dem Dateibereich oder von einer URL ab

🚣 Audio Galerie ermöglicht Audioaufnahmen von Teilnehmern

✓ Bestätigung wird verwendet, um vom Nutzer eine explizite Bestätigung zu erhalten, dass ein bestimmter

Abschnitt angesehen/gelesen/etc. wurde

Bildvergleich vergleicht zwei Bilder mit einem Schieberegler

□■ Diagramm stellt Datensätze in einem Diagramm da

Download stellt eine Datei aus dem Dateibereich zum Download bereit

⟨/> Embed bindet externe Inhalte wie Videos, Grafiken oder Musik ein

> externe Inhalte (IFrame) kann andere Webseiten einbinden

Forum kann Bereiche aus dem Forum direkt in Courseware einbinden

Freitext enthält Text und Bilder, wird mit dem WYSIWYG-Editor bearbeitet

**Galerie** zeigt Bilder aus einem Ordner im Dateibereich an

■ Gruppierung vereint Blöcke und erzeugen eine zusätzliche Navigation → wird durch Container-Layout ersetzt

Interactive Video spielt ein mit Interaktionen angereichertes Video ab, Vips Aufgaben, Text-Einblendungen und

Haltepunkte sind möglich

 $\wp$  Kommentare & Diskussion stellt eine einfache Diskussionsmöglichkeit mit Kurzbeiträgen und Kommentaren bereit ightarrow nach

Anpassungen von Blubber kann ist dieser Block nicht mehr notwendig

Leinwand ermöglicht das zeichnen und schreiben auf einem Bild

Lernkarten bietet Karten zum Umdrehen an, bei denen sich auf beiden Seiten Bild und Test darstellen lässt

**L**ink erstellt einen Link innerhalb der Courseware oder auf eine andere Seite. Beim kopieren, importieren

oder umsortieren bleibt die Funktionalität der internen Links erhalten.

• Merksatz erzeugt einen Merksatz mit Icon und Rahmen

bereitgestellt

🖳 PDF Vorschau zeigt eine PDF-Datei aus dem Dateibereich an

Quellcode lässt die Eingabe und Highlighting von Quellcode zu

? Quiz enthält Aufgaben aus dem Vips-Plugin → wird in Zukunft durch Vips-Plugin bereitgestellt

Schreibmaschine erzeugt den Text Zeichen für Zeichen

Suche stellt eine Suche innerhalb der Courseware zur Verfügung  $\rightarrow$  wird durch Suche in der Sidebar ersetzt

2 Termin zeigt einen Termin oder Countdown an

► Video spielt ein Video aus dem Dateibereich oder von einer URL ab, verschiedene Videotypen und

Qualitäten werden unterstützt

Elementbeschreibung
Zeigt den Beschreibungstext aus den Metadaten eines Strukturelements an

Hausaufgabenordner Ermöglicht Teilnehmern den Upload von Dateien in den Dateibereich

Gruppenordner Analog zu Gruppenordner aus dem Dateibereich

### Geplante Inhaltselemente

konditionaler Ordner Ist mit einem Ordner aus dem Dateibereich verbunden, Sichtbarkeit im Dateibereich ist abhängig von

der Sichtbarkeit in der inhaltsbasierten Ansicht

■ Video-Galerie ermöglicht Videoaufnahmen von Teilnehmern

◆ Titel Zeigt den lang Title eines Strukturelements an, wahlweise als reiner Text oder als Bild oder

Farbverlauf mit Text und optionaler Hintergrundfarbe