Roadmap 2021 - 25	XML Hands-On XSLT & co.				Node Server	X
					JS	X
					XSLT	X
					Django	X
	Erstellt:	05.04.2021	**Freigabe:**	01.12.2021	MongoDB	X

e-Learning mit Tektur *LCMS*

Status:	DRAFT	**Revision:**	1
Autor:	Alex Düsel	XML- / CCMS-Entwickler	

Motivation

Zur *Technischen Dokumentation* gibt es in den Branchen Luftfahrt, Maschinenbau und Automotive spezielle Software, die eine *topicbasierte*, *teilautomatisierte* und *wiederverwendbare* Dokumentation von Bauteilen ermöglicht.

Dabei können Betriebsanleitungen mit mehreren Tausend Seiten entstehen, die für verschiedene Ausgabestrecken automatisch gesetzt werden. Diese sog. *Component Content Management Systeme (CCMS)* sind mit sehr hohen Einführungs- und Betriebskosten verbunden und werden seit mehr als 20 Jahren als Desktopsoftware mit Serverkomponente für Windows OS entwickelt.

Mit aktueller Cloud- / Webtechnologie und Open-Source Software ist dieser Anwendungsbereich aber mittlerweile auch komplett webbasiert abbildbar. Es gibt jedoch weltweit noch sehr wenige Anbieter, die konsequent webbasiert arbeiten.

Der *CCMS* Markt ist in Deutschland zwischen einer Handvoll größerer Anbietern aufgeteilt, die mittels spezialisierter Inhaltsmodelle ihre Kunden seit Jahrzehnten an sich gebunden haben. Deren Systeme sind aber teilweise technologisch veraltet, so dass mittelfristig eine Neuorientierung stattfinden wird.

Es würde den Rahmen von *Tektur CMS* sprengen einen vollständigen Ersatz für die existierende Software zu bieten, da in der Technischen Dokumentation eine Vielzahl unterstützender Software verwendet wird, wie z.B. für das Terminologiemanagement, Übersetzungsmanagement, Bildbearbeitung, Workflowsysteme für Review- und Freigabezyklen, Content Delivery Portale, etc. Diese Schnittstellen müssten ebenfalls bedient werden.

Was aber möglich wäre, und was in der aktuellen Zeit sicherlich für viele Unternehmen einen Mehrwert darstellen würde, wäre sich auf den Bereich e *-Learnings* als führendes Ausgabeformat mit einer *Learning CMS Software (LCMS)* zu spezialisieren.

Konzept und Technologische Orientierung

- Interaktive *e-Learning* Module stellen in der heutigen Arbeitswelt mit Homeoffice und *Lifelong Learning* einen essentiellen Bestandteil im Wissensmanagement in der Industrie dar. Auch im Bereich *Edutainment* und nicht zuletzt im universitären oder Schulbetrieb wären automatisierte Prozesse, mit denen man schnell und unkompliziert interaktive Lernmodule mit definierten Schnittstellen zu anderen Systemen erstellen könnte, vorteilhaft.
- Dabei stellt die Wiederverwendbarkeit von möglichst feingranular zerlegten und einmal übersetzten Lerninhalten eine zentrale Herausforderung dar.

- Hier kann man sich die Konzepte aus der Technischen Dokumentation zu Nutze machen. Dabei wird mittels wiederverwendbarer *XML Bausteine*, die nach dem *Single-Sourcing* Prinzip referenziert und nicht kopiert werden, in verschiedenene Ausgabeformate publiziert.
- Im Bereich *e-Learning* wäre das führende Ausgabeformat ein *autonomes, Interaktives HTML5 Format*, das vom Lernenden heruntergeladen werden kann, oder aber auch Online mit Zusatzfunktionalität ausgeführt werden kann.
- Zusätzlich sollte es noch ein herkömmliches PDF Format geben, welches die Lerninhalte ohne Interaktion zur Archivierung aufbereitet und verschiedene Statistiken zur späteren Auswertung bereithält.
- Die Quelldaten würden dabei medienneutral und zukunftssicher in einem gängigen XML Format, wie z.B. *DITA (Darwin Information Typing Architecture)*, verwaltet.
- Um die Erstellung der Lerninhalte einem *Content Management* und einem kollaborativen *Prüf- und Freigabezyklus* zu unterziehen, sollten die Lerninhalte über eine *cloudbasierte Webapplikation* eingepflegt werden können, die auch die gängigen Funktionen eines *CCMS Systems*, wie *Versionierung*, *Modularisierung*, *Variantensteuerung*, *Rechte- und Rollenmanagement*, *Workflow*, etc. bereitstellt.

Aktueller Bestand

Zum 05.04.2021 kann man mit *Tektur CMS* Inhalte nach dem *DITA* Inhaltsmodell webbasiert erfassen, und diese mittels gängiger Portalfunktionalitäten in verschiedenen Ausgabeformaten (HTML5, PDF, RTF, DITA), Papierformaten (A4, A5) und Layouts konfigurierbar auf einer Website publizieren.

Zusätzlich besteht die Möglichkeit einzelne Topics einem Prüf- und Freigabezyklus zu unterziehen. Dabei können Lektoren und Prüfer Diskussionen an einzelnen Textpassagen anbringen und diese schließlich freigeben, so dass der verantwortliche Redakteur abgestimmte Änderungen an der Publikation vornehmen kann.

Features 2022

Im Jahr 2022 soll *Tektur CMS* zu einem DITA System für Gelegenheitsredakteure ausgebaut werden. Wichtige noch fehlende Funktionalitäten sollen ergänzt werden und bestehende geschliffen werden, so dass Ende 2022 eine kleine, aber feine Lösung vorhanden ist.

Nr	Titel	Beschreibung	Notizen	Erledigt
202201	Weitere Importformate AsciiDoc und ein Exportformat eines kommerziellen CCMS.	Zusätzlich zu den bereits vorhanden Formaten Markdown, Word und Tektur DITA sollen noch weitere Formate importiert werden können.	Das Format AsciiDoc kann über das Opensource Tool AsciiDoctor realisiert werden. Das kommerzielle Exportformat kann natürlich nur nach Absprache mit dem Hersteller importiert werden.	☐
202202	Weitere Ausgabeformate CHM, EclipseHelp, Markdown, Normalized DITA	Neben den existierenden Formaten PDF, HTML5, HTML Paket und DITA XML sollen noch unkompliziert weitere Formate angeboten werden.	Die weiteren Formate sollen über das DITA Open Toolkit integriert werden. Ggf. können auch die OT Formate PDF und HTML eingebunden werden, sollen aber die Tektur eigenen HTML und PDF Formate nicht verdrängen.	☐

Nr	Titel	Beschreibung	Notizen	Erledigt
202203	Diffing mit Vorgängerevisionen	Derzeit können Vorgängerrevisionen eines Topics, Tasks, Memos oder einer Map nur angezeigt werden, aber nicht verglichen werden.	Da in Tektur jedes Element mit einer eindeutigen ID ausgezeichnet ist, kann ein einfacher Diffing Algorithmus angewendet werden, der ausgehend von der ID geänderte, neu hinzugekommene oder gelöschte Texte und Grafiken markieren kann.	☐
202204	Night Mode im HTML5 Format	Im öffentlichen Frontend des Portals soll es einen Umschalter für einen augenschonenden "Nachteulen"-Modus geben.	Der Nachtmodus soll mittels CSS Variablen umgesetzt werden. Dabei soll auch das CSS optimiert werden.	☐

Features 2023

Im Jahr 2023 soll *Tektur CMS* weiter geschliffen werden. Insbesondere soll es weitere Konfigurationsmöglichkeiten bzgl. der Ausgabeformate geben. Dazu wird der vorhandene Layouter-Dialog weiter ausgebaut und auch die Möglichkeit zum Online Bearbeiten einer umfangreichen Konfigurationsdatei gegeben werden.

Nr	Titel	Beschreibung	Notizen	Erledigt
202301	Element- und Strukturgültigkeiten	Im Kapitelstruktureditor sollen über ein Dropdown zuvor definierte Gültigkeiten an den Topics und Tasks gesetzt werden können. Im DITA Content sollen Gültigkeiten über das Lazy-Tag Feature funktionieren.	Die Gültigkeiten sollen in einem Reiter im Map-Dialog definiert werden können.	☐
202302	"Wussten Sie schon?"-Dialog beim Applikationsstart	Beim Start der Webanwendung soll ein optionaler Dialog angezeigt werden, der jeden Tag einen neuen Tipp vorschlägt. Dazu müssen die Tipps gesammelt werden, ggf. mit dem Memo-Editor.	Der bestehende "Willkommen" Dialog soll um diese Funktion erweitert werden.	☐

Nr	Titel	Beschreibung	Notizen	Erledigt
202304	Sprechende Dateinamen für die Export Formate.	Derzeit sind die Dateinamen der Exportformate willkürlich bezeichnet, bspw. "result.pdf" für das PDF Format. Der Dateiname soll den Titel des Informationsobjekts enthalten, sowie die Revisionsnummer und das Exportdatum / Zeit.	--	☐
202305	Erweiterte Konfigurationsmöglichkeit für die Ausgabeformate.	Neben dem bestehenden Layouterdialog soll es auch ein Konfigurationsfile geben, das mittels XML strukturiert ist und online bearbetet werden kann.	Die Konfigurationsdatei soll global für alle Formate der Tekturinstallation greifen. Die Funktionalität ist nicht pro User oder pro Informationsobjekt vorgesehen.	☐
202306	Übersetzungssteuerung einbauen	Momentan muss man jede Sprache für jede Map, Topic, Task oder Memo Objekte separat anlegen. Das soll automatisiert werden.	Die "translate" Funktionen werden analog zu den "review" Funktionen eingebaut. Wenn ein Objekt im Workflow "translate" ist, dann erscheint es beim ausgwälhlten Translator in einem separaten Listing und ist für alle anderen Funktionen gesperrt.	☐

Features 2024

Im Jahr 2024 soll *Tektur CMS* in die Cloud und an einem Subscription Preis-Modell gearbeitet werden - Stichwort Software-as-a-Service.

Nr	Titel	Beschreibung	Notizen	Erledigt
202401	Lokale Cloud-Umgebung aufsetzen	Um ordentlich Entwickeln, Testen und Debuggen zu können, ist es notwendig auf lokalen Rechnern eine Cloud aufzusetzen, bevor man produktiv geht.	Das AWS Localstack Projekt soll hierzu verwendet werden.	☐
202402	Tektur-Instanzen per Admin-Interface anlegen und verwalten.	Ein Administrator kann neue Tektur-Instanzen in der Cloud komfortabel anlegen und verwalten.	Das Admin-interface von *Tektur CCMS* ist mittels Python/Django realisiert, während die Basisapplikation mittels NodeJS realisiert ist. Hier sollen weitere Node-Knoten per Django initialisiert, eingeschaltet, ausgeschaltet und gelöscht werden können.	☐
202403	Skalierungs- und Performanzoptionen in NodeJS	NodeJS ist für ein SaaS-Modell prädestiniert. Hier gilt es die verschiedenen Optionen zu erforschen und anzuwenden, so dass das System optimal läuft.	--	☐
202404	Realtime Features	NodeJS kann Nachrichten an die angeschlossenen Browser pushen. Das ermöglicht eine Vielzahl von Spezialfeatures, die mit herkömmlicher Webentwicklung nicht so einfach umzusetzen sind.	Je nach zeitlicher Kapazität sind die folgenden Funktionalitäten vorstellbar: – Schreib-Lock für Informationsobjekte in Echtzeit aktualisieren - für eine bessere kollaborative Zusammenarbeit der Redakteure – Kommentare und DIskussionen in Echtzeit einblenden – Chats pro Dokument und/oder pro Arbeitsgruppe	☐
202405	Protokollierungs und Mailingfunktionen	Useraktionen sollen umfangreich protokolliert werden und Benachrichtigungsemails sollen verschickt werden.	Die zugehörigen Informationen sollen per Django Admin-Interface verwaltet werden.	☐

Features 2025

Im Jahr 2025 sollen die Arbeiten an der ersten Version von *Tektur CMS* abgeschlossen werden und der Fokus auf das Learning CMS gesetzt werden. Da man davon ausgehen kann, dass in 5 Jahren die Browsertechnologie weiter fortgeschritten ist, soll insbesondere auch nochmal das Thema Browserkompatibilität des XML Editors aufgerollt werden.

Nr	Titel	Beschreibung	Notizen	Erledigt
202501	Learning Schema entwickeln	Um die Eingabe der Elemente einer Learning-Anwendung eingeben zu können, muss zuerst ein Schema erarbeitet werden.	Als Grundlage für das Schema soll der Use Case "Ich möchte interaktiv XML Programmierung Lernen" herangezogen werden. Dabei sollen die bereits bestehenden Testpublikationen auf **www.stylesheet-entwicklung** weiter ausgebaut und professionalisiert werden.	☐
202502	Memo-Editor zum Learning-Editor ausbauen	Das Learning-Schema soll in den Memo Editor möglichst generisch integriert werden	--	☐
202503	XML Editor browserkompatibel machen	Momentan funktioniert der DITA XML Editor nur mit CHROME und Edge. Der DITA XML Editor soll auch fit für den Firefox gemacht werden.	--	☐
202504	Grafikverwaltung	Es soll eine bestehende Open Source Grafik-Verwaltung eingebaut werden.	Hierzu gibt es bestehende Python/Django Apllikationen, die man integrieren kann.	☐
202505	Learning Ausgabeformat	Als Grundlage für das Learning Ausgabeformat soll das bestehende HTML Paket genommen werden, welches um interaktive Learning-Inhalte ergänzt wird.	Es wird unterschieden, zwischen Inhalten, die eine Online-Interaktion benötigen und interaktiven Inhalten, die auch lokal ausgeführt werden können. Die Online-Interaktion sollte nur optional hinzugeschaltet werden, sobald eine Internetverbindung des Users vorliegt.	☐

Geschäftliches

Tektur CCMS ist ein Freizeitprojekt von IT Beratung Alex Düsel (http://www.publiziere.de)
Anregungen und Fragen senden Sie bitte gerne an tekturcms@gmail.com

Copyright 2016 - 2025, Alex Düsel.

All rights reserved.