# le-tex transpect

transpect ist ein Framework für die Prüfung und Konvertierung von XML-Dateien und XML-basierten Dateiformaten wie OOXML (.docx), IDML und EPUB.

## Motivation

Verlage möchten medienübergreifend produzieren. Neben PDF-Output sollen bei möglichst allen Satztiteln EPUB-Daten und evtl. auch XML gemäß Verlagsschema herauskommen. Ein Ansatz ist XML first: zunächst den Input nach XML konvertieren und dann alle Formate daraus generieren. Leider gibt es bei diesem Ansatz praktische Schwierigkeiten, wie z.B.:

* Wie kommt man möglichst automatisch aus heterogenem Input (meistens jedoch Word) zu sauberem XML?[[1]](#footnote-1)
* Kann man Autoren, technische Hersteller und Copy-Editoren, die in Word arbeiten, mit Selbstbedienungstools unterstützen, die Feedback auf dem Weg zur perfekt strukturierbaren Worddatei geben (Prüfung) und vielleicht gleich das XML erzeugen (Konvertierung)?
* Unzureichende XML-Unterstützung von InDesign: Wie geht man mit Autoren um, die einen Umbruch in finaler Anmutung erwarten, bevor sie intensiv korrekturlesen? Bzw. was bedeutet es für den Umbruch? Trägt man die unvermeidlichen Korrekturen ins XML ein und erzeugt und umbricht die InDesign-Datei neu? Lässt man den Setzer neben dem Text und dem Layout das mitgeführte Tagging korrigieren, was ja leider bei InDesign ein Eigenleben neben dem Layout führen kann? Wie stellt man bei mitgeführtem XML sicher, dass z.B. eine Kursivierung auch im Tagging entfernt wurde und nicht unverhofft im erzeugten EPUB wieder auftaucht? Überhaupt: wie lassen sich stark strukturierte Daten wie Literaturangaben oder Inhaltsklassifizierungen als XML mitführen? Das alles verursacht [Kosten und Qualitätsprobleme](http://publishinggeekly.com/2010/11/xml-first-workflows-in-indesign-trugerische-sicherheit/?lang=de).
* Wenn man auf Grund der Probleme bei der InDesign-Produktion auf XML first verzichtet: Wie stellt man sicher, dass beim Export kein Inhalt verloren geht oder verdoppelt wird, dass das Tagging den Verlagsrichtlinien entspricht, und wie erhält man EPUBs, die den Verlags- und Distributorenanforderungen genügen?
* Kann man, ähnlich wie dem Word-Bearbeiter, dem InDesign-Setzer ein Selbstbedienungstool an die Hand geben, das mögliche Fehler wie unverankerte Bilder, unverlinkte Literaturverweise oder händisch formatierte Überschriften (die dann nicht im EPUB-Inhaltsverzeichnis auftauchen) anzeigt und gleich auch noch Verlags-XML erzeugt, validiert und daraus ein EPUB erzeugt?
* Wie geht man mit abweichenden Satzanweisungen, Formatvorlagenbenennungen und Strukturmerkmalen um? Muss man alle Daten über einen Kamm scheren, was Prüf- und Konvertierregeln angeht, oder kann man diese Regeln pro Imprint, Reihe oder gar im Notfall pro Einzeltitel anpassen?
* Wie kann man möglichst die bewährten Satzbetriebe oder In-House-Kräfte weiterbeschäftigen, ohne dass sie Experten in XML und EPUB werden müssen?
* Wie verteilt man Prüf- und Konvertiertools? Interaktive Plugins für Word und InDesign erscheinen attraktiv, bereiten aber Probleme bei der Installation (Rechte), Kompatibilität (Updates der Anwendungen, Wechselwirkung mit anderen Plugins) und bei der Aktualisierung der Prüf- und Konvertierregeln.

Unsere Antwort auf diese Fragen lautet le-tex transpect.

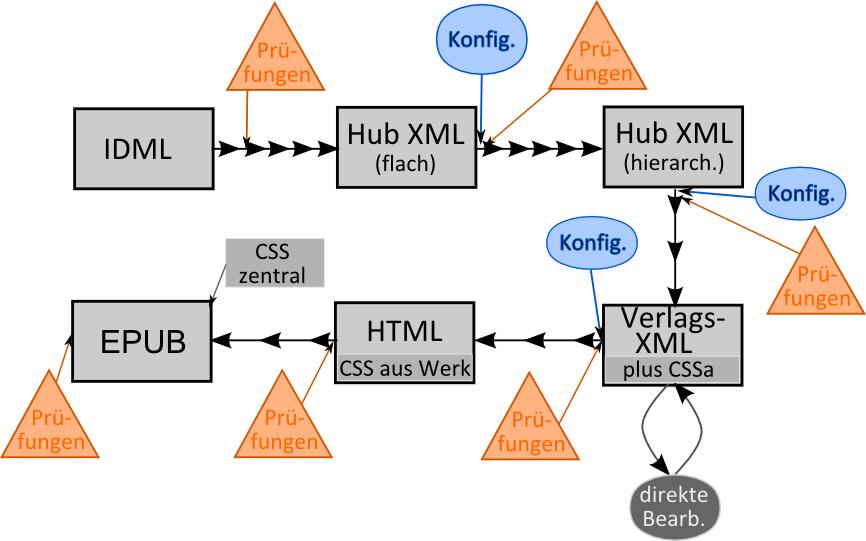


Abbildung Schematischer Ablauf einer transpect-Konvertierungspipeline

### Lösung

transpect bietet

* eine Infrastruktur zur Konvertierung aus XML und aus XML-basierten Dateiformaten wie IDML (InDesign) und OOXML (.docx) sowie zur Generierung von IDML, HTML+CSS, EPUB, .docx und anderer XML-Dialekte;
* Konverter in Standard-Technologien: [XSLT 2](http://www.w3.org/TR/xslt20/) zur XML-Transformation und [XProc](http://www.w3.org/TR/xproc/) zur „Orchestrierung“ mitunter komplexer Konvertier- und Prüf-Pipelines;
* neutrales Zwischenformat („[Hub XML](https://github.com/gimsieke/Hub)“), in dem die Layout-Eigenschaften als [CSS-Properties in XML-Attributen](http://archive.xmlprague.cz/2013/presentations/Conveying_Layout_Information_with_CSSa/CSSa_xmlprague_gimsieke.html#/step-1) (CSSa) dargestellt werden. Diese Repräsentation in CSS-Vokabular hat u.a. den Vorteil, dass Ad-Hoc-Formatierung (z.B. Tabellen¬zellen¬hintergrund¬farbe, Abstand nach Absatz) selektiv über die Prozesskette bis ins EPUB transportiert werden kann. Das erspart aufwändiges Mapping von sporadisch auftretender Spezialformatierung ins Verlags-XML und von dort ins EPUB. Außerdem erlaubt es tiefgehende Schematron-Prüfungen, wie z.B. die Kompatibilität des EPUB mit Kindle oder das Vorhandensein von normalen Absätzen, die wie Überschriften formatiert sind.

Hub-XML-Dateien können so flach sein, wie sie aus Word oder InDesign exportiert werden (alle Absätze, Tabellen etc. auf einer Ebene), oder sie können hierarchisiert sein. Mit hierarchisieren meinen wir nicht nur die Herstellung einer Gliederungshierarchie, sondern auch Listenerkennung, Gruppierung von Abbildungen/Tabellen mit ihren Legenden, Literaturstrukturierung etc, wie es bei so gut wie allen Verlags-XML-Dialekten verlangt wird.

In diesen Konvertierschritt von flach nach wertvoll fließen die meisten Konfigurationsinformationen ein. Die Konvertierung von hierarchisiertem Hub (welches im Wesentlichen DocBook 5.1 ist) in andere XML-Dialekte wie TEI oder JATS/BITS ist dann vergleichsweise einfach. Es wäre jedoch aufwändig, die komplexen Hochkonvertierungen für alle diese Formate separat einzurichten, so dass wir hierfür das neutrale Hub-Format auf DocBook-Basis entwickelt haben, innerhalb dessen wir hochkonvertieren.

* Schema- und Schematron-Prüfungen, die sich an verschiedenen Punkte einer Konvertierpipeline einklinken lassen und deren Fehler- und Warnmeldungen in einer HTML-Ansicht des konvertierten Dokuments am Ort des Fehlers angezeigt werden. Neben den klassischen Schema-Validierungen (incl. DTD) kommt Schematron eine bedeutende Rolle zu, weil man damit viel spezifischer prüfen kann. So mag eine IDML-Datei gegen das IDML-Schema validieren, aber nur mit zusätzlichen Schematron-Regeln erkennt man unverankerte Textrahmen. Die Überprüfung, ob nur Formatvorlagen verwendet werden, die in einem Referenztemplate vorkommen, gelingt ebenfalls mit Schematron. Oder man kann heuristisch eine Warnung ausgeben, wenn eine Überschrift mit „Inhalt“ beginnt, aber wie eine normale Kapitelüberschrift und nicht als IHV-Überschrift ausgezeichnet ist. Solche Warnungen haben sich auch bewährt, um manuelle Trennungen oder harte Zeilenumbrüche in Tabellenzellen oder Legenden zu erkennen und mit einer Warnung zu versehen.
* Kaskadierende Konfiguration: man kann Prüf- und Konvertierregeln global, pro Imprint, pro Reihe und pro Werk hinterlegen; die jeweils spezifischste wird ausgewählt. Typischerweise erben spezifische Konfigurationen die nächst generelleren Regeln und überschreiben sie selektiv. Diese vierstufige Konfigurationskaskade ist dabei nicht in Stein gemeißelt, sondern erweiter- oder reduzierbar.
* Konfiguriert wird das System mit XSLT-, XProc-, Schematron-, Schema- und sonstigen XML-Dateien. Es gibt also kein Administrations-GUI. transpect ist keine Standardsoftware mit einheitlicher Oberfläche. Es ist vielmehr ein Framework (Sammlung von Softwarebibliotheken und Methodologien), mit dem sich schnell auf Kundenbedürfnisse angepasste Konverter erstellen lassen.

Im Zusammenspiel mit anderen Tools (z.B. le-tex aspect oder anderen Web-Anwendungen) kann transpect durchaus eine Administrationsoberfläche erhalten, mit der man Projekte, Nutzerrechte, Rollen und Regeln konfiguriert, aber dies sind eher Bestandteile einer umfassenderen Content-Management- oder Workflow-Lösung. Im Lieferumfang des Prüf- und Konvertier-Frameworks transpect ist derlei nicht enthalten.

* Die Nutzer bekommen in der Regel ein einfaches Web-Formular zum Upload ihrer Dateien bzw. einen überwachten Ordner als WebDAV-Netzwerklaufwerk. Aber auch Aufruf per Kommandozeile ist möglich (Voraussetzung: Java 1.6 oder neuer). Auf Wunsch kann das Prüf- und Konvertiertool auch als ausführbare Datei zum Drag&Drop bereitgestellt werden (gegen Mehrkosten). Wir empfehlen aber die Benutzung als Webdienst, weil so die neuesten Prüf- und Konvertierregeln jederzeit aktuell vorgehalten werden können.
* transpect ist integrierbar in das verlagsweite Content Management. Falls es dafür noch keine Lösung gibt, kann ein Versionskontrollsystem (Subversion) als Content Repository installiert werden, in das die Nutzer mit frei verfügbaren Tools Daten einspielen und von dort auschecken können. Beim Einspielen können automatisch die Tests und Konvertierungen, z.B. in Kunden-XML und EPUB, erfolgen, und fehlerhafte Dateien (Prüfung mit Fehlern, Dateiname entspricht nicht Konventionen, falsches Dateiformat, …) werden zurückgewiesen.

### Leistungen

* Basis-Tools: Open Source
* Beratung und Systemintegration: je nach Umfang. Typischerweise geht mit einem solchen Projekt die Systematisierung der Formatvorlagennamen einher. Evtl. auch die Definition oder Weiterentwicklung eines Verlags-Schemas bzw. die Auswahl und ggf. Anpassung eines existierenden verbreiteten Schemas, wie DocBook, TEI, JATS/BITS, Hinterlegung von Standard-CSS für EPUBs, ggf. Einführung eines Content Repository; Schulungen. Größenordnung: ca. 4–30 Personentage.
* Laufender Support (bei problematischen Daten) und Weiterentwicklung (z.B. neue Schematron-Regel): pro angefangene 5 Minuten.
* Laufender Betrieb auf Hosting- oder Kundenserver (auf Wunsch incl. Versionskontroll-CMS)
* Aufnahme in Continuous Integration-Server: Überwachung, ob Änderungen an den Basiskonvertern zu Problemen bei den installierten Konvertern führen: Jahrespauschale (analog Wartungsgebühr bei proprietärer Software).

Die Basistools werden ständig weiterentwickelt. Der Kunde kann natürlich jederzeit neue Versionen der Open-Source-Software installieren. Vermeintlich unschuldige Änderungen bei Basis-Konvertern zeitigen jedoch manchmal unerwartete Effekte bei bestimmten Konvertierungen. Um das aufzuspüren, werden repräsentative Testdaten aus den Kundenworkflows bei le-tex hinterlegt. Bei jeder Aktualisierung eines Basistools werden die Konvertierungsergebnisse mit der beim Kunden installierten Konfiguration mit den Ergebnissen verglichen, die sich bei Update auf die neuesten Versionen ergäbe. Dies geschieht vollautomatisch. Im Fall von Unterschieden gibt es eine Warnung, und der Kunde kann entscheiden, ob bei seiner Konvertierung etwas an die neuen Tools angepasst werden muss, so dass er updaten kann, oder ob die Konverter bis auf Weiteres unverändert bleiben. Nach Möglichkeit versuchen die le-tex-Entwickler schon im Vorfeld, die Bibliotheken mit den Kundeninstallationen kompatibel weiterzuentwickeln. Das kann aber nur gelingen, wenn die Kundeninstallation und Referenzdaten im Continuous-Integration-System hinterlegt sind.

## Warum Open Source?

Gerade der letzte Punkt im voranstehenden Abschnitt schafft Produktionssicherheit und macht den Unterschied zwischen Konvertierung zu Experimentier- oder Privatzwecken und gesicherter Produktion im industriellen Maßstab aus. Die Weiterentwicklung wird finanziert aus diesen Jahrespauschalen, aus der kostenpflichtigen Umsetzung von Feature Requests der Kunden und aus der Erbringung von Support.

Dies ist das Geschäftsmodell von le-tex transpect.

Die Software wurde unter der permissiven BSD-Lizenz veröffentlicht. Das heißt, dass jeder die Software für eigene Zwecke anpassen kann, sofern er den ursprünglichen Urheber (le-tex) nennt. Im Unterschied zur GPL müssen Änderungen nicht unter derselben Open-Source-Lizenz veröffentlicht werden.

Wir haben mit uns gerungen, ob wir die Software freigeben sollten. Vor allem gab es Bedenken unter dem Gesichtspunkt Fairness: wir haben fünf- bis sechsstellige Beträge in die Entwicklung investiert, und alle Wettbewerber sollen die Früchte unserer Arbeit kostenlos nutzen dürfen? Ebenso kann es Bedenken beim Kunden geben, wenn wir einen Feature Request umsetzen sollen, der später jedem Nutzer zugute kommen wird, und wenn aber nur der Kunde, der zuerst danach gefragt hat, dafür zahlen soll.

Wir denken, beides ist fair: Gerade wenn Wettbewerber die Methodologie übernehmen, rechtfertigt das unseren XSLT-2-/XProc-basierten Batch-Processing-Ansatz im Zoo der verfügbaren Tools. Als Initiator des Produkts kommt le-tex unter den möglichen Dienstleistern eine herausgehoben Stellung zu, und Wettbewerber werden nicht unbedingt jeden Feature Request ihrer Kunden selbst umsetzen wollen, sondern le-tex beauftragen.

Und wenn Kunden es für unfair halten, dass sie die Entwicklung eines Features finanzieren, so sollten sie bedenken, dass andere Kunden und le-tex den Rest der Software zuvor bereits finanziert haben. Und Kunden sollten diese übersichtlichen Entwicklungskosten vergleichen mit Lizenzkosten und sonstigen Einschränkungen proprietärer Software. (Abgesehen davon ist uns keine proprietäre Standardsoftware für derartige Prüf- und Konvertierpipelines bekannt.)

Der Open-Source-Ansatz soll außerdem Bedenken seitens des Kunden adressieren, die sich um die Wartung und Weiterentwicklung von Software sorgen, die ein Dienstleister neben seinem Hauptgeschäft programmiert hat. Selbst wenn le-tex mit ca. 100 Mitarbeitern nicht ganz klein ist: Die Entwickler könnten abspringen, die Firma könnte das Interesse an der Weiterentwicklung verlieren oder ganz aus dem Markt verschwinden, etc. In diesen Fällen (von denen wir nicht ausgehen, aber aus Sicht des Kunden ist die Entscheidung für die Lösung eines kleinen Softwareanbieters natürlich riskanter) kann der Kunde die Lösung selbst weiterpflegen oder einen anderen Anbieter beauftragen. Auch unter diesem Gesichtspunkt hat le-tex ein Interesse daran, dass gerade auch Wettbewerber die Tools nutzen, denn das senkt aus Sicht des Kunden das Investitionsrisiko und erleichtert somit die Entscheidung für eine solche Lösung.

## Module

Die Module besitzen einen unterschiedlichen Reifegrad. Produktiv einsetzbare Module sind in der folgenden Liste nicht speziell gekennzeichnet, während \* ausbaubedürftig bedeutet, und \*\* bedeutet: Modul im Planungsstadium. Weitere Module entstehen in der Regel, wenn ein Kunde ein spezielles Feature benötigt (s.o.).

* IDML→XML-Konverter

Herausziehen von XML-Tagging oder Erzeugung von Hub-Format

* .docx→Hub-XML-Konverter
* IDML-Synthese aus XML \*
* .docx-Synthese aus XML \*
* Kaskadiertes Laden von Konfigurationen
* Darstellung von Prüfungsfehlern (Schematron, Schema/DTD) in HTML-Rendering des Quelldokuments (Schema/DTD: \*)
* Konverter von flachem Hub-XML in hierarchisches Hub-XML
* Hub-XML→DocBook-Konverter
* Hub-XML→TEI-Konverter \*
* Hub-XML→BITS/JATS-Konverter
* XML→HTML-Konverter (aus den o.g. XML-Formaten) mit der Möglichkeit, neben zentralem CSS selektiv die Layouteigenschaften der Quelldateien durchzureichen
* EPUB-Generator (EPUB 2 und 3) mit Unterstützung von custom Fonts, Page Lists, Fixed Layout, Read-Aloud-Synchronisation und Zugriff auf externe Metadaten (die als XML im Dateisystem oder per HTTP verfügbar sind)
* Crossref-Lookups von Literaturzitaten (Voraussetzung: Verlag ist Crossref-Mitglied)

1. Hier ist eine Fußnote [↑](#footnote-ref-1)