Überlegungen und Vorschläge zur Online-Publikation von wissenschaftlichen Journalen

Version 1.0

Heinz Rosenkranz MSc
Februar 2016

Inhaltsverzeichnis

1 Betrachtung eines Referenzprojektes

Als Referenzprojekt wird das „Related Content Service“ des Springer-Verlags betrachtet.
Siehe dazu:

> Springer Nature and UNSILO release pioneering new
> interactive Related Content Service on SpringerLink
> http://www.springer.com/gp/about-springer/media/press-releases/corporate/springer-nature-and-unsilo-release-pioneering-new-interactive-related-content-service-on-springerlink/3406436

Das „Related Contend Service“ des Springer-Verlags bietet Zugang zu 9 Millionen Artikeln. Es handelt sich dabei um eine „Big Data“-Anwendung, die mit entsprechendem Aufwand unterstützt wird, u. a.:

- durch Textmining, Mustererkennung und „natürliche Sprachverarbeitung (NLP)“
  für die Erkennung und Extraktion relevanter Fachausdrücke und Phrasen
- durch Vektorraum-Modelle zur Bestimmung
  der Relevanz bzw. Ähnlichkeit von Dokumenten
- durch Einsatz von Ontologien zum Erkennen von
  Begriffshierarchien und verwandter Begriffe
- durch Interaktive grafische Visualisierung
  des jeweiligen Themenkomplexes

Die Anwendung wurde in Kooperation mit der Firma Unsilo (unsilo.com) aus Dänemark entwickelt, die laut Eigendefinition Spezialist für Semantische Anwendungen ist. Aus den aktuellen Job-Angeboten der Firma Unsilo läßt sich auf die verwendeten Technologien schließen:

- Client side programming (JavaScript, HTML5, CSS, JQuery, AngularJS)
- Server side programming (J2EE, Spring, RESTExpress, Spark)
- Search technology (Lucene, Elasticsearch, SOLR)
- Big Data stores (Cassandra, DynamoDB, Google Datastore, MongoDB)
- Big Data solutions (Hadoop, Spark, Storm, AWS SWF, Kafka)
- Cloud platforms (AWS, Google Cloud, Cloud Foundry)
- Machine Learning, Text Mining, Natural Language Processing, Deep Learning

Zur Anwendung kommen weiters “agile Projektmethoden”, die in regelmäßigen Intervallen die aktive Beteiligung des Auftraggebers einfordern, zum Beispiel für Feedback und User-Experience (UX).

> Das Agile Manifest beschreibt diese interessante Denkschule:
> http://www.agilemanifesto.org/principles.html

1.1 Metadaten und „Related Content“

Von Interesse ist dabei, wie die Journal-Artikel im Kontext einschlägiger Schlagworte (Metadaten) und ähnlicher Literatur (Related Content) eingebettet werden. Die dazu verwendeten Konzepte dienen in erster Linie zur Navigation im verlagsinternen Produktangebot – ob und wie die Metadaten und Dokumentverweise auch externen Diensten (Google, Archive, Bibliotheken) angeboten werden, ist nicht ersichtlich. Für die weitere Betrachtung siehe folgende Referenzseite:

> Agriculture in the Central Asian Bronze Age
> http://link.springer.com/article/10.1007/s10963-015-9087-3

Bild 1: Artikel-Startseite mit Abstract und traditioneller Beschlagwortung (Keywords)

1.1.1 Abstract und Keywords

Jeder Artikel enthält eine vom Autor verfasste Zusammenfassung (Abstract) und ausgewählte Schlagworte (Keywords). Damit wird die traditionelle Volltextsuche gezielt unterstützt und darüber hinaus auch essentielle Metadaten für eine weiterführende semantische Nutzung vorgegeben.

1.1.2 Related Content

Durch rechnergestützte Methoden wie z. B. Textmining, Mustererkennung und „Natural Language Processing (NLP)“ lassen sich aus dem Artikeltext weitere Schlagworte sowie Phrasen und (kognitive) Konzepte extrahieren, die in einem Vektorraum-Model zu einer eindeutigen Dokumentcharakteristik verdichtet werden.

Die nachfolgende Abbildung zeigt Literaturempfehlungen, die auf Basis der erkannten Schlagworte und der daraus berechneten Dokumentähnlichkeit aufgelistet werden.

Bild 2: Relevante Dokumente zu vorgegebenen Schlagworten

Für „Related Content“ ist das Umfeld einer Big-Data Anwendung nützlich, da sowohl die Relevanz eines Schlagwortes wie auch die Relevanz eines Dokumentes (Vektorraum-Modell) nur bei großem Datenbestand sinnvoll ermittelt und genutzt werden kann.

Schlagwort-Relevanz

- groß bei häufiger Termfrequenz im Dokument
- klein bei häufiger Termfrequenz im Dokumentbestand

Dokument-Relevanz bzw. Ähnlichkeit

- hoch bei hoher Übereinstimmung relevanter Terme
- niedrig bei geringer Übereinstimmung relevanter Terme

Die rechnergestützte Bewertung von Begriffs- und Dokumentrelevanz erweitert die geordnete, von Autoren vorgegebene Beschlagwortung, um ein interessantes Phänomen: den „Serendipity“-Effekt, er sorgt dafür, dass sich manchmal Zusammenhänge offenbaren, nach denen man gar nicht gesucht hat.

1.1.3 Semantische Zusammenhänge

Semantische Zusammenhänge bzw. Ähnlichkeiten ergeben sich bei der Einbeziehung von Ontologien, entweder aufgrund der Begriffshierarchien, oder durch explizite assoziative Beziehungen (Synonyme).

Auf der Unsilo-Webseite findet sich dazu ein anschauliches Beispiel:

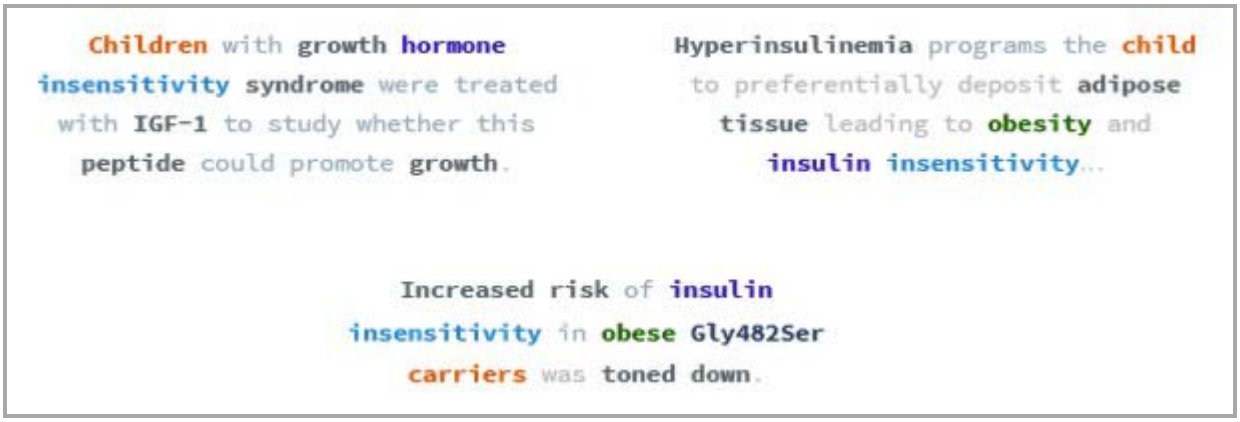


Bild 3: Semantische Zusammenhänge in unähnlichen Textpassagen

Im gezeigten Beispiel ist erkennbar, dass hier Ontologien aus unterschiedlichen Abstraktionsebenen zusammenspielen, zum Beispiel:

Midlevel Ontologies

- Humans (parent, child, sister, relatives …)
- Attributes (grow, tone down, insensitiv, hyper …)

Domainspezific Ontologies

- Medical (carrier, syndrome, obesity, adipose, tissue …)
- Biomedical (hormone, peptide, insulin …)
- Codes (IGF-1, Gly482Ser …)

Der Einsatz von Ontologien, also Begriffsstrukturen für semantische Anwendungen, ist ein hoch anspruchsvolles Unterfangen. Der „Serendipity-Effekt“, also die Offenbarung unerwarteter Zusammenhänge, ist hier aber kein zufälliges Phänomen, sondern Zielsetzung.

1.2 Interaktive Grafik (Relationship Map)

Die Relationship Map unterstützt als interaktive Grafik in besonders effizienter Weise die Suche nach ähnlichen Dokumenten. Hier sind auf dem ersten Blick interessante Zusammenhänge ersichtlich, zum Beispiel Dokument-Cluster (Häufungen) und weit entfernte „Wissensinseln“.

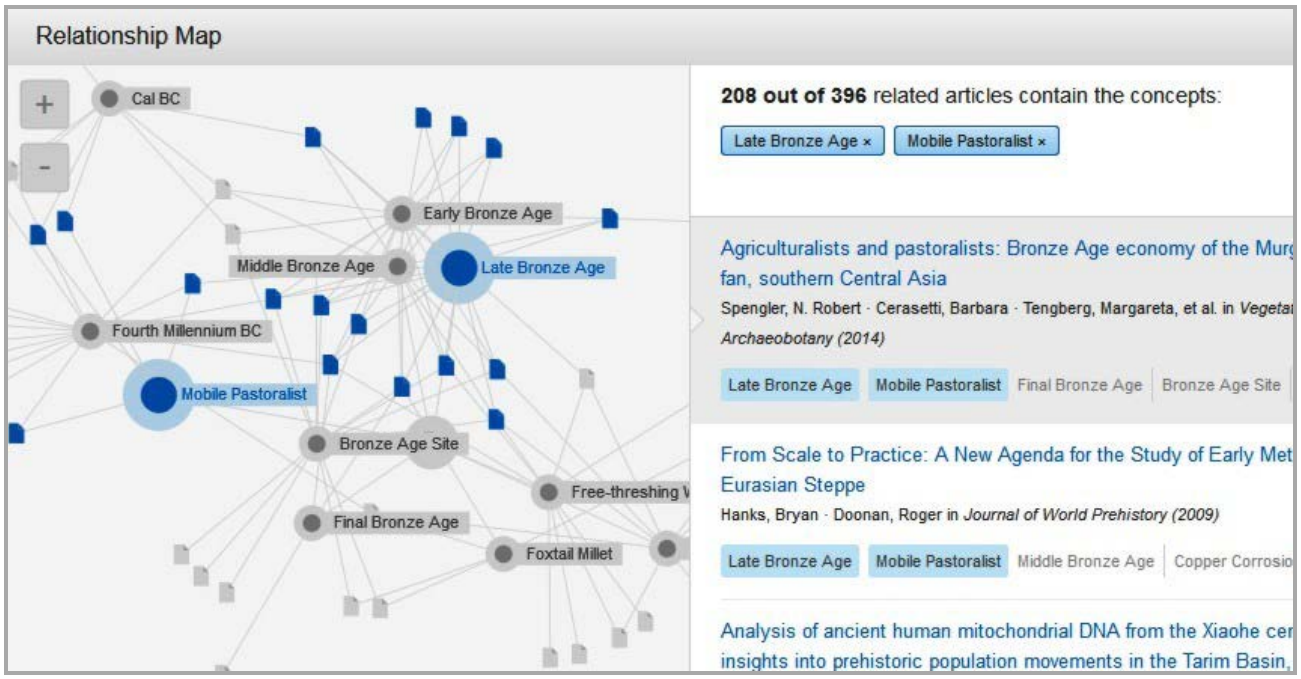


Bild 4: Relationship Map mit Querverweise auf relevante Dokumente

Die Implementierung einer interaktiven Grafik erfolgt üblicherweise als JavaScript-Bibliothek mit dynamisch erzeugtem Datenbestand (Nodes und Edges). Es wird eine HTML 5 Umgebung vorausgesetzt, also Browser der letzten Generation (ab Firefox 20, IE 8).

1.2.1 Anbieter für interaktive Grafiken (JavaScript-Bibliotheken)

Data Visualization Software Lab, Zoomcharts (Litauen)
https://zoomcharts.com/en/
Lifetime License with Premium Support US$ 900.-

GoJS (US)
http://gojs.net/
Developer License for OEM-Products US$ 2.995
Unlimites Domains License US$ 2.795

JGraph, mxGraph (UK)
https://www.jgraph.com/
Lifetime License, 12 month support Euro 7.400

Keyline (UK)
http://keylines.com/
Licensing muss verhandelt werden

1.3 HTML- und PDF-Ansicht

Die Journal-Artikel werden sowohl im druckfreundlichen PDF-Format wie auch im webgerechten HTML-Format bereitgestellt. Das HTML-Format unterscheidet sich inhaltlich vom PDF-Format nur durch die Verlinkung der Literaturverweise.

Siehe dazu folgende Referenzseite:

Domesticating Animals in Africa: Implications of Genetic and Archaeological Findings
http://link.springer.com/article/10.1007/s10963-010-9042-2/fulltext.html

1.4 Fazit

- Für kleine Verlage sind Entwicklungskosten in Millionenhöhe für eine „Big Data"-Lösung keine argumentierbare Option.
- Ein echter Kosten-Nutzenvorteil ist in Anbetracht der hohen Investitionskosten kaum erkennbar. Anders im Rechtsbereich, hier ist z. B. die Information über einschlägige Präzedenzfälle pures Geld wert.
- Eine HTML-Ansicht der Artikel bringt kaum Vorteile, da strenger Review-Auflagen eine freizügige Ergänzung der Artikeln mit Web-Links möglicherweise untersagen? Der Mehraufwand ist ebenfalls erheblich:
  - Konvertierung der PDF-Dateien
  - Medienaufbereitung (Extraktion von Bildern, Diagrammen, komplexen Tabellen)
- Interaktive Grafiken (Relationship Maps) sollten in Betracht gezogen werden, da sie beeindruckend wirken und auch für Digitalisierungsprojekte von Interesse sind.
- Vorschlag: Spezialisierung auf Digitalisierung alter Journale und vergriffene Literatur als Alternative zum „Big-Data"-Pomp?

1.5 Kostengünstige Alternativen

Als kostengünstige Alternative bzw. Vorgehensweise zu der betrachteten Referenz-Anwendung bieten sich mehrere Möglichkeiten an. Im wesentlichen sind das:

1. **Kollaboration mehrerer Verlage**
   Bündelung der Finanzierungsmittel
   Nutzung von vorhandenem Knowhow und Synergien
   Nutzung von Webservices und Open Source Projekten (Apache …?)
   Outsourcen der kostspieligen Entwicklung (Baltikum, Slowenien …?)
2. **Content Management Lösung (CMS)**
   Großer Markt und viele Anbieter
   Starke Produkt- und Anbieterbindung
3. **XML-Lösung**
   Neutraler und flexibler Lösungsansatz
   Auf Scriptsprache basierendes Anwendungsumfeld (einfache Wartung und Adaption)
   Kaum Abhängigkeit von Trends, Technologien und Anbietern

Im Folgenden werden die Möglichkeiten einer XML-Lösung betrachtet.

2 Publikationsquellen

Traditionellen Publikationsquellen, im wesentlichen Buch, Journal oder Lexikon, unterscheiden sich in ihrem Organisationskonzept, der inhaltlichen Struktur und dem Metadaten-Aufkommen. Es ist daher schwierig, ein einheitliches und universelles Konzept für Online-Publikationen zu definieren, das alle Aspekte berücksichtigt und alle Anforderungen erfüllt. Bei allen Lösungsansätzen sollte aber ein wichtiger Faktor im Vordergrund stehen: die Autoren-Akzeptanz. Denn ohne kooperierende Autoren gibt es keine Publikation, die Leser später konsumieren könnte.

Allgemein sind für eine Online-Publikation auch folgende Aspekte zu berücksichtigen:

- **Keine Seitenorientierung – sondern Fließtext**
  - Fußnoten können nicht am Seitenende positioniert werden
  - Index-Einträge können nicht auf Seitennummern referenzieren
  - Ausdruck von fortlaufenden Webseiten ist oft ein Problem
- **Eingebettete Medien (Bilder, Diagramme, Tabellen)**
  - die Positionierung von Medien ist einfacher, da Seitenbegrenzungen entfallen
  - die Größe von Bildern und Diagrammen ist nicht begrenzt, sie können gezoomt werden
  - Copyright-Hinweise müssen evtl. als Watermark vorgesehen werden?
- **Verweise und Zitierregeln**
  - Zitierregeln müssen als Ersatz für Seitenverweise definiert werden
  - für umfangreiche Publikationen ist dafür eine Strategie erforderlich, z. B. Unterteilung in identifizierbare Abschnitte
- **Wiederverwendbarkeit**
  - Artikeln, Kapiteln und Texte sind nicht mehr in eine Publikation „eingebunden“
  - sie können in beliebigem Kontext wiederverwendet werden
  - müssen aber in jedem Kontext Informationen über die Herkunft enthalten

2.1.1 Buch

- Ein Buch ist einer Wissensdomäne und gegebenenfalls als Band einer Serie zugeordnet
- Die Organisation erfolgt in Kapiteln, die ein vorgegebenes Thema stufenweise abhanden
- Qualifizierte Metadaten bzw. vom Autor vorgesehene Schlagworte finden sich üblicherweise in den Register-Anhängen. Die zugehörigen Seitenverweise müssen neu organisiert werden
- Fußnoten bzw. Anmerkungen finden sich üblicherweise am Seitenende oder Kapitelende. In beiden Fällen ist eine Neuorganisation sinnvoll, die den Lesefluß unterstützt
- Tabellen und Diagramme sollten als „Abbildungen“ behandelt und beschrieben werden

Objekt	Identifikation	Metadaten
Wissensdomäne	Verlags-URL/domäne	optional
Serie	… /domäne/serie	optional
Buch	…/buch	**obligat: Verlagsinformation, Autor(en)**
Kapitel	…/buch/kapitel	**obligat: Titel, Schlagworte**
Fußnote		selektiv: Anmerkung, Quellenverweis
Abbildung (Diagramm)	…/kapitel/abb	**obligat: Copyright** optional: Beschreibung
Tabelle	…/kapitel/tab	optional: Beschreibung
Index		selektiv: Personen, Orte, Sachworte
Anhang		?

2.1.2 Journal

- Ein Journal ist einer Wissensdomäne und einer Journal-Serie zugeordnet
- Die Organisation erfolgt in Redaktionelle Beiträge und Artikeln
- Die Artikeln sind unterschiedlich, und verwandte Themen über die Journal-Serie verteilt
- Eine Verlinkung verwandter Themen muss organisiert werden, z. B. mittels Artikel-Index
- Komplexe Tabellen sollten als „Abbildungen" behandelt und beschrieben werden
- Formeln können mittels MathML konstruiert werden

Objekt	Identifikation	Metadaten
Wissensdomäne	Verlags-URL/domäne	optional
Journal-Serie	… /domäne/serie	**obligat; Artikel-Index**
Journal	…/journal-Ausgabe	**obligat: Verlagsinformation**
Redaktion. Beitrag		optional
Artikel	…/journal-Ausgabe/artikel	**obligat: Titel, Schlagworte, Autor(en)**
Abstract	…/artikel/abstract	**obligat**
Fußnote		selektiv: Quellenverweise
Abbildung (Diagramm)	…/artikel/abb	**obligat: Copyright** optional: Beschreibung
Tabelle	…/artikel/tab	optional: Beschreibung
Formeln	…/artikel/mat	MathML

2.1.3 Lexikon und Chronologie

Lexika und Chronologien sind als Mischformen zu sehen. Sie enthalten neben Einträgen oft auch allgemeine Kapitel, und die Einträge basieren auf unterschiedlichen Konzepten, zum Beispiel:

- alphabetisch geordnete Einträge zu bestimmten Themen
  Beispiel: Motif-Index, ÖBL, ML
- chronologisch geordnete Einträge zu verschiedenen Themen
  Beispiel: Ministerratsprotokolle, Tagebücher
- einfach strukturierte Einträge (kompakte Zeilenstruktur)
  Beispiel: LGB, Mundarten-Lexikon
- stark strukturierte Einträge (Absatzstruktur, eingebettete Medien)
  Beispiel: Deutsche Inschriften

Objekt	Identifikation	Metadaten
Wissensdomäne	Verlags-URL/domäne	optional
Lexikon	…/lexikon	**obligat: Verlagsinformation** optional Autor(en)
Eintrag	…/lexikon/eintrag	**obligat: Lemma** optional: Schlagworte, Autor selektiv: Quellenverweise
Abbildung (Diagramm)	…/eintrag/abb	**obligat: Copyright** optional: Beschreibung
Tabelle	…/eintrag/tab	optional: Beschreibung

2.2 Fazit

- Die Vielzahl an Publikationsquellen macht es schwer, ein einheitliches Konzept für die Erstellung und Präsentation von Online-Publikationen zu finden
- Jede Publikation setzt andere Schwerpunkte - entweder tiefgehende Kapitelstruktur, reichhaltige Textformatierung oder hoher Medienanteil (Bilder, Diagramme, Tabellen)
- Publikationen sind von Natur aus nicht seitenorientiert, sondern thematisch in die Tiefe (Buch) oder Breite (Lexikon) orientiert
- Ein erfolgreiches Publikationskonzept setzt die Akzeptanz der Autoren voraus,
  - denn Autoren müssen jeden Tag mit dem vorgesehenen Werkzeug arbeiten
  - für eine zufriedene Leserschaft reicht ein einmaliger Design-Aufwand
- Hilfreich ist hier, immer die semantische Perspektive im Vordergrund zu sehen: Also den Inhalt (Bedeutung) einer Publikation, und nicht das Aussehen. Es hilft einem Piloten nicht, wenn das Notfall-Manual nur ein schönes Seitenlayout besitzt
- Jedes abgeschlossene Thema kann als eigenständiges „Wissensobjekt“ verstanden werden
- Bei „Wissensobjekten“ geht es nicht nur um die Lokalisierung als „Related Content“, sondern auch um die Wiederverwendbarkeit der Information (Re-Usability)
- Die Wiederverwendbarkeit von Informationen bzw. Publikationen ist ein wichtiger Beitrag zur Eindämmung der „Informationsflut“

3 XML-Lösung allgemein

3.1 Warum XML?

Eine berechtigte Frage. Auf den ersten Blick, insbesonders für Außenstehende, erscheint XML äußerst unattraktiv – zumindest im Vergleich mit WYSIWYG-Konzepten wie zum Beispiel Word, wo das Aussehen des Dokumentes immer präsent ist. Einen gelegentlichen Autor wird man kaum von den Vorzügen von XML überzeugen können – sogar Techniker würden eher zu LaTex tendieren.

Anders die Situation im Verlagswesen bzw. bei hauptberuflichen Autoren und Redakteuren, wo die Erstellung regelkonformer Dokumente und Publikationen im Mittelpunkt steht. Hier wurde bereits durch SGML, der Vorgängersprache von XML, umfassende Aufbauarbeit und Motivation geleistet. Trotzdem ist es interessant, die allgemeinen Vorzüge von XML zu kennen.

3.1.1 Mantra der XML-Vorzüge

XML bedeuted e**X**tensible **M**arkup **L**anguage, auf deutsch "Erweiterbare Auszeichnungssprache". Diese Begriffserklärung ist etwas irreführend, da sie die Bedeutung von XML nicht im vollen Umfang wiedergibt. Eine Antwort auf die Frage nach Sinn und Bedeutung von XML ist vielschichtig und hängt von Niveau und Intention der Fragestellung ab. Einem interessierten Laien reicht die Antwort:

> XML ist vergleichbar mit HTML, aber wesentlich flexibler.

Autoren, Programmierer oder Systemarchitekten sind mit dieser knappen Aussage nicht zufrieden. Sie haben unterschiedliche Ansprüche und erwarten sich konkrete Vorteile beim Einsatz von XML.

XML: Allgemeine Aspekte

- XML ist reiner Text und somit für Menschen lesbar, angemessen verständlich, und (zur Not) mit einfachen Texteditoren bearbeitbar
- XML ist als reiner Text ideal für Langzeitarchivierung geeignet
- XML ist durch die verbindliche Markup-Struktur ideal für Maschinen lesbar. „Wellformedness“ und „Validität“ sind dabei wichtige Eigenschaften
- Transaktionen finden immer mehr zwischen Maschinen statt, Floskeln und Höflichkeiten werden durch Metadaten und semantisches Markup abgelöst

XML: Formale Aspekte

- XML ist ein defacto-Standard (W3C-Recommendation)
- XML ist zugleich Syntax, Metasprache und Sprachfamilie
- Als Syntax ist XML eine einfache Notation mit wenig Regeln und Konventionen
- Als Meta-Sprache ist XML ein Werkzeug (bzw. eine Spezifikation) für die Erzeugung von konkreten Markup-Sprachen (XML-Dialekte bzw. XML-Anwendungen)
- Als Sprachfamilie umfaßt XML neben einigen Kernsprachen (XSLT, XPath …) unzählige Dialekte für praktisch jeden denkbaren Anwendungsfall (e-Business, e-Government, e-Justiz ...) und Dokumentstandards wie DocBook, JATS, TEI …

XML: Technische Aspekte

- XML ist plattform-neutral (reiner Text) und dank Unicode international
- XML ist präsentationsneutral und wird erst bei Bedarf in ein gewünschtes Format transformiert, zum Beispiel in anzeigbares HTML oder ausdruckbares PDF
- XML ist objektorientiert (DOM). Diese Eigenschaft entfaltet sich aber erst zur Laufzeit im Arbeitsspeicher und ist im „serialisierten“ Textdokument nicht erkennbar

XML für Autoren

- XML ist in besonderem Maße für Autoren und Web-Designer interessant, weil es strikt zwischen Inhalt, Struktur und Präsentation unterscheidet.
- Durch bedeutungstragendes Markup sind Dokumente "selbsterklärendend"
- Ein XML-Dokument kann mehrfach genutzt werden, z. B. als Buch, als Webseite, oder als Objekt in unterschiedlichem Kontext (Re-Usability)
- XML-Dokumente sind im Schnitt um 30 - 50 % schlanker als vergleichbare HTML-Dokumente, da nur der Dokumentinhalt und nicht die Darstellung beschrieben wird
- Praktisch alle aktuellen Web-Browser unterstützen XML und XSLT
- Ein XML-Dokumente ist durch seine definierter Struktur auf formale Vollständigkeit und Richtigkeit überprüfbar (validierbar)

XML für Software-Entwickler

- Weit über 90% der XML-Anwendungen finden sich als Datenschnittstellen im IT-Bereich. Viele Programmierer haben daher XML-Knowhow und Kompetenz
- Das Dokument Objekt Model (DOM) ist als verbindlicher Standard eine Schnittstelle zu JavaScript und Hochsprachen wie Java und C#.
- Ereignisgesteuerte Parser (SAX) für die Verarbeitung großer Dokumente und Datenströme sind als Alternative zum komfortablen DOM verfügbar
- XSLT selbst ist eine mächtige Scriptsprache, die durch Java/Javascript-Binding auch mit zusätzliche Funktionen aus externen Sprachen erweitert werden kann
- XML-Dokumente brauchen nicht unbedingt ein Anwenderprogramm, sie schaffen sich die Anwendung durch XSLT-Transformation selbst (smarte Dokumente)

XML für Software-Architekten

- XML ist als flexibles und plattformunabhängiges Format gleichermaßen gut geeignet für Archivierung, Transport, Verarbeitung und Darstellung von Dokumenten und Daten
- XML ist das Bindeglied zwischen Datenwelten bzw. heterogenen Serverarchitekturen
- Service orientierte Architekturen (SOA, EAI) und Webservices basieren auf XML
- Behördenanwendungen (z. B. Digitale Signatur) sind ohne standardisierte (kanonisierbare) XML-Dokumente undenkbar
- Das semantische Web als kommenden Herausforderungen basiert auf XML-Standards wie RDF/S, OWL und Topic Maps

3.2 XML-Vokabular

Die Wahl einer geeigneten XML-Anwendungssprache ist für Autoren bzw. Redakteure von zentraler Bedeutung, denn sie sind unmittelbar damit konfrontiert. Der spätere Nutzer einer Web-Publikation sieht (und bezahlt) nur das marktübliche Ergebnis, nicht aber den Erstellungs-Aufwand.

Praktisch alle bekannten Dokument-Standards verwenden reichlich Vokabular, einsteils um alle denkbaren Szenarien abzudecken, andernteils wegen aufwendiger Detail-Modellierung, z. B.:

Verschachtelte Elemente	statt Attribute kurz und bündig
<contrib contrib-type="author"> <name> <surname>Forster</surname> <given-names>Anne</given-names> </name> <role>Research physiotherapist</role> <aff> Department of Health Care </aff> </contrib>	<Autor Name="Forster" Vorname="Anne"> Research physiotherapist | Department of Health Care </Autor>

Diese extensive Modellierung ist ein Relikt aus der Frühzeit von XML, bzw. von SGML vererbt. Hier wird das Akronym eXtensible Markup Language, also die Freizügigkeit eines Vokabulars, zu wörtlich genommen, und die eigentlichen Vorzüge und Intentionen der XML-Sprache ignoriert:

- XML unterstützt in erster Linie die minimalistische Modellierung eines Vokabulars, die wahre Mächtigkeit des Vokabulars liegt in der Fokussierung auf die Anwendung
- Bei minimalistischer Modellierung dominiert der Inhalt und nicht das Markup
- Minimalistische Modellierung bevorzugt Attribute statt Elemente:
  - Elemente konstituieren in erster Linie die Dokumentstruktur
  - Attribute transportieren die Vielzahl an Metadaten und strukturlosen Informationen
- Der Erfolg von HTML als Weltsprache des Web basiert auf einem minimalistischen Konzept, erst durch CSSS mit seinen hunderten Styles wird HTML zu einem mächtigen Werkzeug

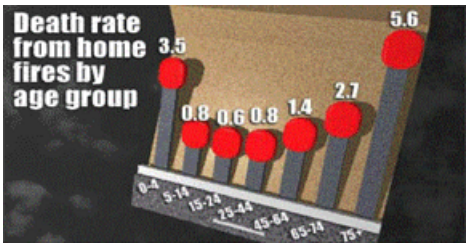
3.2.1 Minimalistisches XML-Vokabular

Minimalistisches XML-Vokabular wird für den jeweiligen Anwendungsfall maßgeschneidert und ist somit besonders autorenfreundlich. Dabei geht es aber nicht nur um die Befindlichkeit von Autoren, sondern zum Beispiel auch um die Effizienz bei Massendigitalisierung.

Minimalistisches XML-Vokabular enthält im Schnitt 20-30 intuitive Vokabular-Konzepte und nutzt die nativen Möglichkeiten von XML sowie die Mächtigkeit von XSLT, zum Beispiel:

- Reihenfolge, Hierarchie und Position der XML-Elemente
- Optionalität von Attributen, sowie Defaultwerte und vorgegebene Enumerationen
- Generische Elemente, die durch semantische Flexibilität vielfach nutzbar sind, zum Beispiel: <p type="simple">, <p type="footnote">, <p type="cite"> usw.
- Verwendung von bekannten HTML-Tags für Textformatierung (p, span,b, i, br, sup, sub …)
- Nutzung von Processing-Instructions, zum Beispiel für Fußnotenverweise und ähnliches
- Einfache Textstrukturen im CSV-Format, die durch XSLT-Stringfunktionen aufgelöst werden, zum Beispiel bei Namensangaben „Familienname | Pseudonym | Künstlername“ oder „Ortsname | historischer Name“

3.2.2 Komplexe Inhalte (Tabellen, Diagramme, Formeln)

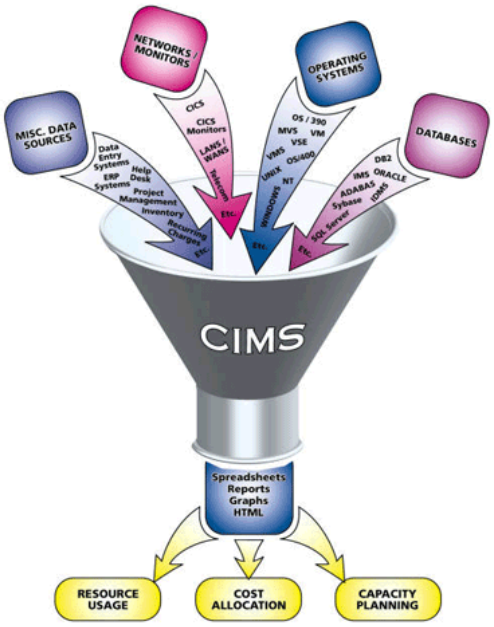


Komplexe Inhalte wie Tabellen, Diagramme und Formeln sind in der Regel visuelle Ergänzungen zum geschriebenen Text. Aus Autorensicht sind sie nützlich, oder sogar unverzichtbar – denn sie erlauben im Sinne der bekannten Metapher „Ein Bild sagt mehr als tausend Worte“ die Sicht auf Zusammenhänge, die sich aus geschriebenem Text nicht erschließt.

Grundsätzlich lassen sich komplexe Inhalte durch XML-Vokabular konstruieren, zum Beispiel durch HTML-Tables, skalierbarer Vektorgrafik (SVG) und MathML. Das zugehörige XML-Vokabular ist aber kaum autorenfreundlich. Die nachfolgende Formel zum Beispiel erfordert weit über zwanzig MathML-Kodezeilen:

$$f(a) = \frac{1}{2\pi i} \oint_\gamma \frac{f(z)}{z-a} dz$$

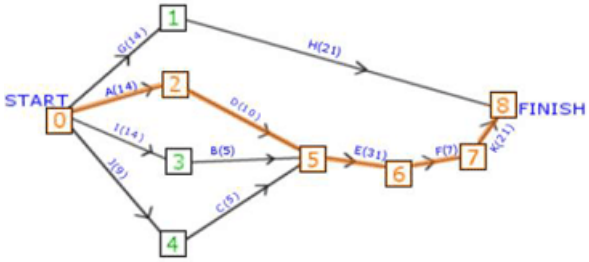
Solche komplexe Konstrukte sind keine Zielsetzung für XML-Dokumente, da Bruchstriche oder Wurzelsymbole nichts zur inhaltlichen Beschreibung (Semantik) beitragen. Um eine Gleichung allgemein zu verstehen, also Ähnlichkeiten zu anderen Gleichungen zu finden, oder sogar Schlußfolgerungen daraus ziehen zu können, muss man wissen bzw. mitteilen, ob sie zum Beispiel auf Naturgesetzen basiert, oder empirisch gewonnen wurde, oder probabilistischer Natur ist, und ob sie eine lineare, quadratische, exponentielle oder partielle Form hat usw.



Hier wäre es durchaus sinnvoll, im Sinne der Bild-Metapher komplexe Inhalte als Abbildungen einzufügen, zum Beispiel eine handskizzierte statt einer konstruierten Gleichung, und ergänzend zu beschreiben – denn „ein Bild sagt mehr als 1000 Worte“ ist in der semantischen Welt nicht gültig.

Das eigentliche Problem ist aber möglicherweise gar nicht die Komplexität des Vokabulars, sondern die grundsätzliche Einschränkung der gestalterischen Möglichkeiten des Autors. Mit geeigneten Werkzeugen lassen sich bequem Tabellen und Visualisierungen erzeugen, die weit über das Machbare einer XML-Anwendung hinausgehen. Die nachfolgend angeführte Webseite zeigt zum Beispiel an die hundert Diagrammarten, die für wissenschaftliche Publikationen von Bedeutung sind:

Periodic Table of Visualization Methods:
http://www.visual-literacy.org/periodic_table/periodic_table.html



3.3 XSLT-Transformationen

In XML-Dokumenten wird oft viel Aufwand vergeudet, um Aussehen (Layout) und Funktionalität zu beschreiben, mangels Kenntnis der mächtigen Transformationssprache XSLT. XML-Dokumente sollten aber grundsätzlich auf Inhalt und Metadaten fokussiert sein, und nicht irgendeine Publikationsform bevorzugen. Die Darstellung von XML ist Aufgabe von XSLT.

3.3.1 HTML-Erzeugung

Die XML-HTML Transformation ist eine der wichtigsten XSLT-Anwendungen zur Erzeugung von Publikationen. Sie basiert im wesentlichen auf zwei Konzepten:

XPath
Mit XPath-Anweisungen werden gezielt Element- oder Atribut-Inhalte aus dem Quelldokument gelesen. XPath-Anweisungen sind vergleichbar mit SQL-Statements für relationale Datenbanken. Sie können hochkomplex formuliert und konditioniert werden, z. B.:

```
<xsl:value-of select="count(//topic[(.//@href='See') or (.//@href='Stausee')])"/>
```

XSL-Templates
Templates sind Schablonen, z. B. für HTML-Konstrukte, die mit Inhalten des Quelldokumentes befüllt werden. Hier sind für einen HTML-Programmierer keine Grenzen gesetzt – es können beliebige CSSS-Styles und Javascript-Funktionen inkludiert werden, z. B.:

```
<!-- Match Footnote Processing-Instruction -->
Hier wird ein Fußnotenverweis als hochgestellter Text formattiert
und zugleich der Fußnotentext als Hoverhelp zugewiesen.
<xsl:template match="processing-instruction()[name() = 'nn']">
  <xsl:param name="nr" select="normalize-space(.)"/>
  <sup style="color:#800000; cursor:hand;" title=" {//p[@nr = $nr]}">
    <xsl:value-of select="$nr"/>
  </sup>
</xsl:template>
```

3.3.2 XML-XML Transformation

Eine der wichtigsten Eigenschaften von XML ist die Wellformedness. Sie garantiert, dass sich der gesamte Dokumentinhalt wohldefiniert innerhalb von Elementen oder Atttributwerten befindet. Somit ist sichergestellt, dass jedes Dokument-Detail mittels XPath extrahiert werden kann, und in weiterer Folge in eine neue XML- Struktur eingebettet werden kann.

Damit ist es möglich, aus einem vorhandenen XML-Modell ein anderes zu erzeugen, zum Beispiel aus einem JATS-Dokument ein TEI-Dokument.

3.3.3 PDF-Erzeugung mittels XSL-FO

Diese relativ aufwendige und anspruchsvolle Möglichkeit wird vorerst nicht weiter betrachtet, da eher die umgekehrte Vorgehensweise von Bedeutung ist – also die XML-Erzeugung aus vorgegebenen Formaten wie Word und PDF.

3.4 Fazit

Warum XML?

- Gelegentliche Autoren wird man kaum von XML überzeugen können
- Weit über 90% der XML-Anwendungen sind „Datenschnittstellen“.
  IT-Experten haben daher hohe XML-Kompetenz und das erforderliche Knowhow,
  aber kaum Praxiserfahrung mit anspruchsvollen Textdokumenten
- XML ist keine beliebige Sprache oder Format, sondern eher ein Paradigma,
  mit überraschend vielen Facetten und Vorzügen
- XSLT ist eine mächtige Scriptsprache, die aus „abstrakten“ XML-Dokumenten
  konkrete Repräsentationen macht
- Wenn man von IT-Experten XML-Lösungen einfordert,
  wird man als Gesprächspartner auf Augenhöhe behandelt

XML-Vokabular

- Extensive Modellierung ist ein unnötiges Relikt aus der Frühzeit von XML
- XML unterstützt in erster Linie die minimalistische Modellierung eines Vokabulars,
  die wahre Mächtigkeit des Vokabulars liegt in der Fokussierung auf die Anwendung
- Bei minimalistischer Modellierung dominiert der Inhalt und nicht das Markup
- Minimalistisches XML-Vokabular nutzt die nativen Möglichkeiten von XML
  sowie die Mächtigkeit von XSLT

Komplexe Inhalte (Tabellen, Diagramme, Formeln)

- Komplexe Inhalte lassen sich im Prinzip mit XML-Vokabular beschreiben,
  allerdings nur mit hohem Kodieraufwand und vermutlich geringer Autorenakzeptanz
- Ein Bild sagt mehr als 1000 Worte – diese Metapher ist in der semantischen Welt nicht gültig.
  Bilder sind visuelle Ergänzungen des Gesagten und sollten ausführlich beschrieben werden
- Ein Bild sagt mehr als 1000 Worte – diesmal im positiven Sinne: Diagramme und Tabellen sind
  visuelle Ergänzungen des Gesagten, man sollte sie nicht durch „1000 Striche“ und unhandliches
  XML-Vokabular rekonstruieren, sondern einfach als Abbildung erzeugen und beschreiben

XSLT-Transformationen

- In XML-Dokumenten wird oft viel Aufwand vergeudet, um Aussehen und Funktionalität
  zu beschreiben, mangels Kenntnis der mächtigen Transformationssprache XSLT
- XML-Dokumente sind präsentationsneutral und auf Inhalt und Metadaten fokussiert
- XSLT-Transformationen sorgen dafür, dass jede Präsentationsform optimal unterstützt wird,
  z. B. PDF-Druckseiten, HTML-Webseiten oder Graphen (z. B. Relationship-Maps)
- XSLT erlaubt auch die Konvertierung in andere XML-Dialekte (z. B. JATS in TEI)

4 XML Anwendungssprachen für Publikationen

Im wesentlichen kommen folgende Anwendungssprachen bzw. Dokumentstandards für eine nähere Betrachtung in Frage:

- **Objektorientiertes XML-Konzept**
  Ein für jeden Anwendungsfall „maßgeschneidertes" Konzept, basierend auf minimalistischen XML-Vokabular, Index-Dateien (Linked Data) und zugehörigen Stylesheets für Darstellung und gegebenenfalls Transformation in andere XML-Anwendungssprachen (JATS, TEI …)

- **XML Topic Maps**
  Der ISO-Standard XML Topic Maps ist ein universelles Dokumentmodell für Wissens-repräsentationen (Ontologien). Topic Maps könnte als etablierter ISO-Standard somit eine interessante Alternative zwischen frei erfundenem „minimalistischen Vokabular" und den „überspezifizierten" Dokumentstandards wie DocBook, JATS, TEI usw. sein.

- **Journal Article Tag Suite JATS/BITS**
  Journal Article Tag Suite (JATS) ist ein vom National Center for Biotechnology Information entwickelter Dokumentstandard, der für den Austausch und die Archivierung wissenschaft-licher Publikationen benutzt wird. Auf JATS basierend wurde auch ein Standard für wissen-schaftliche Bücher (BITS) entworfen.

- **Text Encoding Initiative**
  TEI ist ein von der gleichnamigen Organisation TEI entwickelter Dokumentstandard, der vorzugsweise in der Geisteswissenschaften verwendet wird.

  Wird bei Bedarf im Detail betrachtet

- **DocBook**
  DocBook ist ein Dokumentstandard, der sich besonders zur Erstellung von Büchern, Artikeln und Dokumentationen im technischen Umfeld (Hardware oder Software) eignet. DocBook ist ein offener Standard, der von der Organization for the Advancement of Structured Information Standards (OASIS) gepflegt wird.

  Wird bei Bedarf im Detail betrachtet

- **Darwin Information Technology Architecture DITA**
  Wird nicht im Detail betrachtet.

4.1 Objektorientiertes XML-Konzept

Im einfachsten Fall sieht ein objektorientiertes Konzept vor, Publikationen in ihre elementaren Teile zu zerlegen und als XML-Objekte einzeln identifizierbar zu machen, zum Beispiel:

- ein Buch wird in eigenständige Kapitel-Objekte zerlegt
- ein Journal wird in eigenständige Artikel-Objekte zerlegt,
  die Redaktionellen Beiträge in einem eigenen Objekt zusammengefasst.
- Ein Lexikon oder eine Chronologie wird in Eintrags-Objekte zerlegt,
  einfach strukturierte Einträge können entweder
  - unter einem alphabetischen Eintrag (A, B, C ...) zusammengefasst werden
  - oder einer Periode zugeordnet werden.

Die so gewonnenen Objekte sind über eine hierarchisch aufgebaute URI eindeutig identifizierbar und können bei Bedarf auch als Ensembles in einen anderen Kontext präsentiert werden, zum Beispiel als Journal-Artikeln zu einem Themenschwerpunkt. Diese Wiederverwendbarkeit (Re-Usability) ist ein wichtiger Aspekt in Anbetracht der allseits kritisierten Informationsflut.

4.1.1 Verzeichnisstruktur

Beispiel einer einfachen Verzeichnisstruktur, die zugleich Basis für eine hierarchische Identifikation ist:

Die jeweiligen **Band-Verzeichnisse** der Ministerratsprotokolle enthalten die einzelnen Kapitel- oder Protokoll-Objekte, im darunter liegenden PDF-Verzeichnis finden sich die zugehörigen PDF-Druckversionen.

Das gemeinsame **_styles-Verzeichnis** enthält die zentralen XSLT-Stylesheet, die Layout, Verlagsinformation und HTML-Funktionalität für alle Objekte bereitstellen.

4.1.2 Index-Datei (Linked Data ?)

Die Index-Datei, die jeder Publikation zugeordnet wird, erfüllt eine wichtige Funktion:

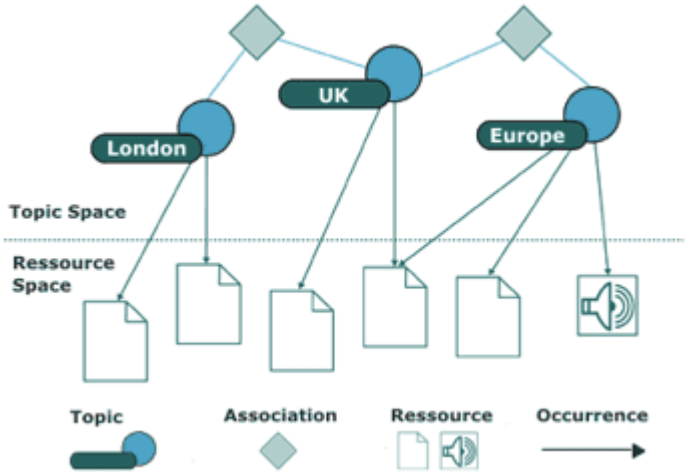
- im einfachsten Fall bildet sie ein Verzeichnis (Index) für die Objekte,
  dies gilt insbesonders auch für reine PDF-Publikationen, um ein Minimum an Navigation und Metadaten bereit zu stellen.
- im Normalfall enthält sie zu jedem Objekt eine definierte Klassifikation
  - und erlaubt damit eine thematische Zuordnung des Objektes
  - den Export von Metadaten aus dem jeweiligen Objekt (OAI-PMH ?)
  - und die Konstruktion einer graphischen Darstellung (Relationship Map)
- bei Bedarf kann sie auch zusätzliche Metadaten enthalten und damit erweiterte Ansprüche abdecken. Insbesonders durch Einsatz von XML Topic Maps lassen sich anspruchsvolle Konzepte realisieren (Linked Data ?)

Die Index-Datei ist ein XML-Dokument, das in Kombination mit einem geeigneten Stylesheet in eine beliebige Form transformiert werden kann, zum Beispiel:

- in andere XML-Formate, die für Archiv- und Bibliotheksdienste gewünscht werden
- in eine Liste von SQL-Statements für Relationale Datenbanken
- in ein CSV-Format (Comma-Separated Values) für Webservices

4.2 XML Topic Maps (ISO 13250)

Der ISO-Standard XML Topic Maps ist ein universelles Dokumentmodell für Wissensrepräsentationen (Ontologien).



Dieses Modell ist nicht nur ein etablierter Standard für die Beschreibung von Themen und deren Beziehungen (Topics und Associations), sondern auch ein vielseitiges Konzept mit asymptotisch mächtiger Ausdrucksfähigkeit. Ausgehend von einfachen, generischen Elementen läßt sich durch hinzugefügte Semantik ein beliebig komplexes Anwendungsmodell konstruieren.

Ein wichtiger Aspekt ist die Möglichkeit, strukturierte Informationen in die Topic Maps direkt einzubetten, anstatt auf externe Ressourcen (Occurrences) zu verweisen. Damit lassen sich aus an sich abstrakten Wissenslandkarten konkrete Wissensdokumente erzeugen.

Das TAO, also die „Weisheit“ von Topic Maps mit seinen Topics, Associations und Occurrences, basiert auf knapp 20 Elemente, die rasch erlernbar und einfach anwendbar sind. Topic Maps als etablierter ISO-Standard könnte somit eine interessante Alternative zwischen frei erfundenem „minimalistischen Vokabular“ und den „überspezifizierten“ Dokumentstandards wie DocBook, JATS, TEI usw. sein.

In der Praxis zeigt sich aber leider, dass dieser seit dem Jahr 2000 existierende „europäische“ Standard vom US-Markt und dem einflußreichen W3C-Konsortium weitgehend ignoriert wird.

4.3 JATS (Journal Articel Tag Suite)

Die Journal Article Tag Suite (JATS) ist ein vom National Center for Biotechnology entwickelter Dokumentstandard, der für Austausch und die Archivierung wissenschaftlicher Publikationen benutzt wird. Auf JATS basierend wurde auch ein Standard für wissenschaftliche Bücher (BITS) entworfen.

Das JATS-Konzept sieht drei Modelle vor:
Siehe dazu auch http://jats.nlm.nih.gov/index.html

- **Journal Archiving and Interchange** is the most permissive of the Tag Set. It defines elements and attributes that describe the content and metadata of journal articles, including research and non-research articles, letters, editorials, and book and product reviews.
- **Journal Publishing** is a moderately prescriptive Tag Set, optimized for the archives who wish to regularize and control their content.
- **Article Authoring** is the most prescriptive of the Tag Sets. The Article Authoring Tag Set is optimized for authorship of new journal articles, where regularization and control of content is important, and where it is useful to have only one way to tag a structure.

Konkret verwendet JATS in der aktuellen Version 272 Elemente und 137 Attribute.

Für eine effiziente Anwendung von JATS ist eine Aufbereitung bzw. Adaptierung des offiziellen Standards (Journal Publishing Tag Set) sinnvoll, die folgenden Schritte umfasst:

- Bereitstellung einer kompakten DTD (simpleJATS)
- Erstellung eines JATS-Beispieldokumentes als Leitfaden (Beispiel im Beispiel)
- Erstellung eines XSLT-Stylesheets für HTML-Transformation

Ob JATS dann direkt für Autorentätigkeit verwendet wird, oder mittels XSL-Transformation aus einem minimalistischen Vokabular erzeugt wird, ist wahlweise möglich.

4.3.1 Bereitstellung einer kompakten DTD (simpleJATS)

Die offizielle Journal Article Tag Suite (JATS) ist eine DTD (Document Type Definition), deren Konstituenten unübersichtlich und etwas verwirrend auf 40 Dateien verteilt sind. Insbesonders die häufige Verwendung von (an sich nützlichen) Entities zur Formulierung gemeinsamer Eigenschaften erschwert das Verständnis, da diese Entities häufig in einer Datei definiert, in einer anderen Datei spezifisch ergänzt, und erst in einer dritten Datei instanziert (genutzt) werden. Für effizientes Arbeiten ist daher die Extraktion des relevanten Regelwerkes aus der Tag Suite sinnvoll.

Die Extraktion liefert folgendes Ergebnis:

- Eine übersichtliche Darstellung des Regelwerkes als Basis
  (defacto Pflichtenheft) für stabile XSLT-Stylesheets
- Eine kompakte Datei mit weniger als einem Zehntel der Suite-Bibliotheksgröße
  – also rasche Ladezeit bei Validierung und bei Nutzung in kontextsensitiven Editoren
- Eine vorsichtige Einschränkung bzw. Auslegung (Konvention) des Regelwerkes
  - mächtig genug, um auch fremde JATS-Dokumente (z. B. PNAS) zu verstehen
  - modernen Anforderungen angepasst (z. B. E-Mail-Kontakt statt Postadresse, FAX)
  - defensive Konvention bei der Verwendung von rekursiven Konzepten
  - restriktive Konvention bei der Verwendung von alternativen Möglichkeiten

Konventionen ergänzen das strikte Regelwerk der simpleJATS-DTD. Die Einhaltung der Konventionen ist nicht zwingend (z. B. durch fremde JATS-Dokumente), die abweichenden Regeln werden aber im XSLT-Stylesheet möglicherweise nicht berücksichtigt.

4.3.2 JATS-Beispieldokument als Leitfaden (Beispiel im Beispiel)

Wissenschaftliche Artikel sind keine Prosa oder Lyrik, die man einfach zeilenweise schreibt, sondern eine komplexe Ansammlung von Mikrotypographie, Tabellen, Formeln, Diagrammen und Verweisen.

Diese Komplexität lässt sich durch kein Autorenwerkzeug umfassend abdecken. Eine WYSIWYG-Umgebung wie zum Beispiel Word ist für Autoren praktisch sinnlos, da weder Layout, Textformate, eingebettete Grafiken noch Fußnoten nutzbar sind. Und für mathematische Ausdrücke und Metadaten-Anreicherung ist selbst Word zu schwach. Autoren werden also in irgend einer Form mit XML-Quelltext konfrontiert werden müssen, zum Beispiel mit vorkonfektionierten Dokumentvorlagen und Beispieldokumenten. Anstelle einer WYSIWYG-Umgebung wird ein gängiger Browser verwendet, der das in einem XML-Editor bearbeitete Dokument unmittelbar als HTML-Ergebnis anzeigt.

Die Autorenakzeptanz sollte kein Problem sein, denn wissenschaftliche Autoren sind seit jeher mit dem Leidensdruck unzureichender Autorenwerkzeuge konfrontiert. Im Nachfolgenden wird anhand des JATS-Dokumentmodells aufgezeigt, welche Anforderungen von Autoren bedient werden müssen, bzw. welche Unterstützung durch einen JATS-Leitfaden erforderlich ist.

Das JATS-Dokumentmodell weist vier Hauptelemente auf:

- **front** zwingender Dokumentteil mit Journal- und Artikel-Metadaten sowie Abstract
- **body?** optionaler Dokumentteil mit Artikel-Inhalt
- **back?** optionaler Dokumentteil mit Fußnoten und Literaturverweisen
- **floating-groups?** optionaler Dokumentteil für Abbildungen, Diagramme usw.

front

- **Journal-Metadaten**: trivialer Inhalt, der vorkonfektioniert werden kann
- **Artikel-Metadaten**: trivialer Inhalt, der vom Autor bearbeitet werden muss, u. a. Abstract, Autorenliste, Danksagung und Kategorisierung des Artikels
- **front** kann als einziger Inhalt in einem JATS-Dokument aufscheinen, z. B. wenn für eine Einreichung nur das vorläufige Abstract erforderlich ist

Acknowledgements

We thank Mark Krosky, Katia Koelle, and Kevin Chung for programming and technical assistance. We also thank Drs.

Communicated by Avner Friedman, University of Minnesota, Minneapolis, MN

Journal-Metadata

The National Academy of Sciences
pmc: pnas
pubmed: Proc Natl Acad Sci U S A
publisher: PNAS
issn: 0027-8424

Article-Metadata

publisher-id: 181325198
publisher-id: 3251
doi: 10.1073/pnas.181325198
other: jPNAS.v98.i18.pg10214
pmid: 11517319
Category: Physical Sciences
Applied Mathematics
Category: Biological Sciences
Genetics

body

- komplexer Inhalt, hier ist hohe Autorenunterstützung erforderlich
- Rekursive und redundante Modellierung, die bei exzessiver Nutzung unkontrollierbare Ergebnisse produziert. Hier muss die Nutzung durch Konvention geregelt werden
- Tabellen lassen sich mit einfachem HTML-Vokabular konstruieren – kein Lernaufwand
- Mathematische Ausdrücke setzen einschlägige Kenntnisse voraus (TEX oder MathML)

back

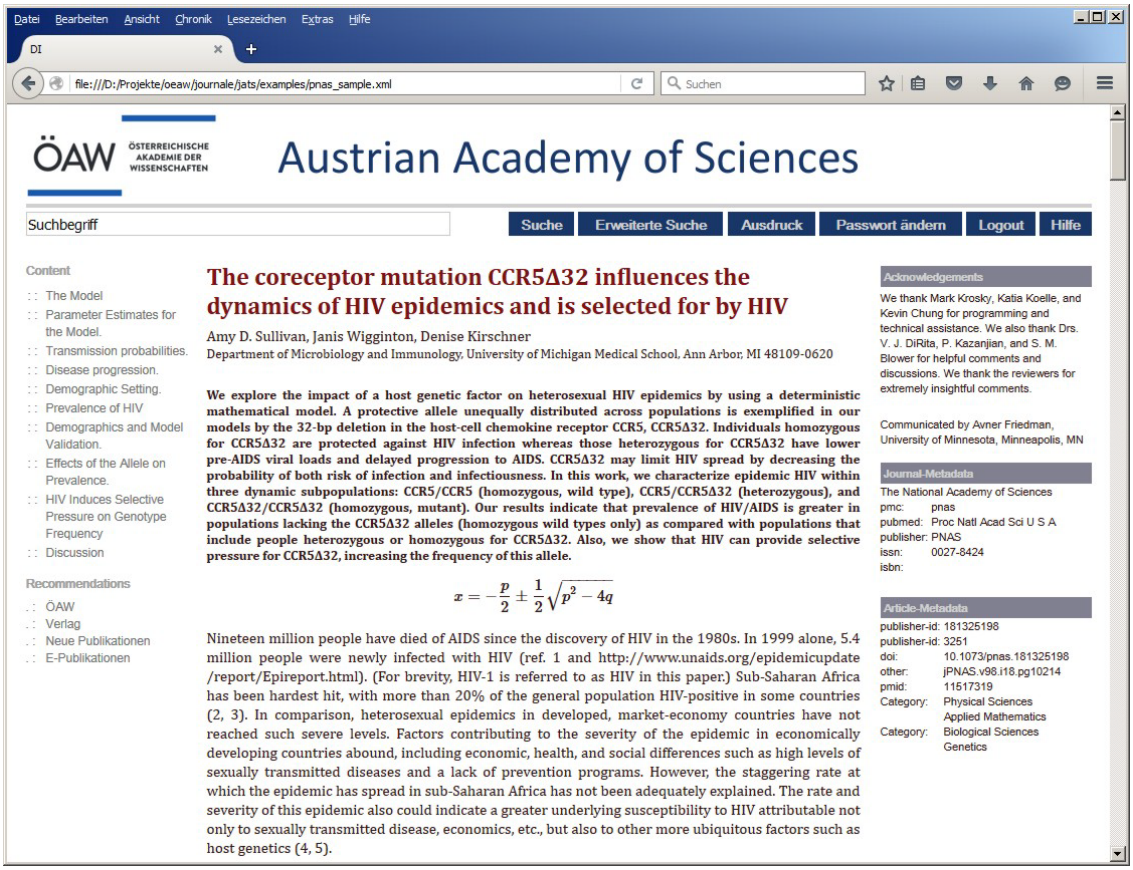
- eher trivialer Inhalt, dient zur Auflistung von Fußnoten und Literaturverweisen
- Querverweise innerhalb des Dokumentes werden auf Korrektheit validiert, hier wird der Autor umfassend unterstützt bzw. diszipliniert

floating-groups

- kaum genutzt, dient bei Bedarf zur Gruppierung des Bildmaterials als Anhang

4.3.3 XSLT-Stylesheet für HTML-Transformation

Für die Darstellung von JATS-Dokumenten wird ein bewährtes Layout verwendet, das bereits bei anderen Publikationen im Einsatz ist (Deutsche Inschriften, Ministerratsprotokolle). Es bietet ein eine Kopfsegment mit freier Titelgrafik und Verlagsfunktionen (Suche, Login ...) sowie ein dreispaltiges Konzept für Artikel-Navigation, Artikel-Inhalt und Metadaten.



Die **Artikel-Navigation** enthält im Wesentlichen:

- die Kapiteltitel für bis zu drei Kapitel-Hierarchien (Sections)
- optionale Auflistung von Tabellen, Diagrammen, Formeln und Abbildungen
- optional Anzahl der Tabellen, Diagrammen, Formeln und Abbildungen
- sowie Links zu Verlags-Empfehlungen

Der **Artikel-Inhalt** enthält im Wesentlichen:

- Artikel-Titel und gegebenenfalls Sub-Titel
- Autorenliste und/oder Affiliation (Nennung der Institutionen)
- ein Abstract
- Inhalt mit bis zu drei Kapitelhierarchien (Sections)
  - erste Hierarchie mit führendem fettem Kapiteltitel
  - zweite Hierarchie mit fettem inline-Kapiteltitel im ersten Absatz
  - dritte Hierarchie mit kursivem inline-Kapiteltitel im ersten Absatz
  - tiefere Kapitelhierarchien werden wie dritte Hierarchie behandelt
- Eine beliebige Mischung aus Absätzen (Paragraphs) und Toplevel-Elementen (Tabellen, Diagramme, Abbildungen und Formeln)
- Absätze können neben typografisch gestaltbarem Text (fett, kursiv ...) auch inline-Grafiken und inline-Formeln enthalten, sowie Verweise
- Absätze können beliebig Toplevel-Elemente enthalten, was zu exzessiv tief geschachtelten Strukturen führen kann
- Eine abschließende Auflistung von Fußnoten und Literaturverweisen

Die **Metadaten** enthalten im Wesentlichen:

- optionales Journal-Cover
- Danksagungen (Acknowledgements)
- Journal-Metadaten
- Artikel-Metadaten
- und optional Related-Content (soferne verfügbar)

Weitere Besonderheiten sind:

Darstellung von Tabellen

Für die Darstellung von Tabellen wird der bekannte HTML-Syntax verwendet, also eine einfache Notation bestehend aus table, tr (table-row) und td (table-data)

Table 1: Children's genotype

Parents		Mother		
Father		W/W	W/Δ32	Δ32/Δ32
	W/W	$X1_{,j}$	$X1_{,j}$ $X2_{,j}$	$X2_{,j}$
	W/Δ32	$X1_{,j}$, $X2_{,j}$	$X1_{,j}$, $X2_{,j}$, $X3_{,j}$	$X2_{,j}$, $X3_{,j}$
	Δ32/Δ32	$X2_{,j}$	$X2_{,j}$, $X3_{,j}$	$X3_{,j}$ $3_{,j}$

Die Gestaltung der Tabelle obliegt demAutor. Er bestimmt bei Bedarf

- ob ein Tabellenrand (border) gezeichnet wird
- wie groß die Zellenabstände sind (cell-spacing)
- und wie groß die Einrückung des Textes ist (cell-padding)

CSS-Styles werden nicht ubterstützt, womit ein wichtiges Gestaltungsmittel entfällt.

Darstellung von mathematischen Ausdrücken

$$f(a) = \frac{1}{2\pi i} \oint_\gamma \frac{f(z)}{z-a} dz$$

Für die Darstellung mathematischer Ausdrücke werden zwei Konzepte unterstützt:

- **tex-math**
  tex-math Ausdrücke werden vom Autor in gängigem TEX/LaTEX formuliert, bzw. von einem entsprechenden Mathematik-Werkzeug erstellt, und in der erzeugten HTML-Seite direkt an eine JavaSript-Bibliothek übergeben, die das typographische Rendern des Ausdrucks durchführt.

  Dies setzt die Verwendung einer open-Source JavaScript-Bibliothek voraus, die bei kommerzieller Anwendung vermutlich eine Lizenzierung erfordert.

- **mathML (XML)**
  mathML Ausdrücke werden vom Autor in einem XML-Vokabular formuliert und in der HTML-Seite direkt vom Browser interpretiert.

  Dies setzt die Verwendung eines aktuellen, HTML 5-fähigen Browsers voraus.

4.3.4 Fazit

- JATS + MathML funktioniert nur mit aktuellen Browsern (HTML5) - das sollte aber kein Problem sein, da praktisch jeder Facebook- oder Youtube-Nutzer schon zur Verwendung aktueller Browser gezwungen wird
- Eine WYSIWYG-Umgebung für Autoren ist praktisch sinnlos, da weder Layout, Textformate, eingebettete Grafiken noch Fußnoten nutzbar sind. Und für mathematische Gleichungen und Metadaten-Anreicherung ist selbst Word zu schwach. Autoren werden also in irgend einer Form mit XML-Quelltext konfrontiert werden müssen
- JATS verwendet kaum enumerierte (vordefinierte) Attributwerte und erlaubt damit den Autoren hohe Freiheit bei der Nutzung (Wildwuchs-Gefahr)
- JATS hat leider kaum semantische Features, bietet aber zu fast allen Elementen optionale Attribute an (specific-use), die man für Semantik nutzen könnte
- Mathematische Ausdrücke sind eine Herausforderung - in allen Dokumentwerkzeugen. Da gibt es keine einfache Lösung, entweder muss TEX/LaTEX oder MathML (XML) verwendet werden. Wissenschaftlichen Autoren sollten aber diese Technik beherrschen
- Es gibt nun eine funktionierende JATS-Umgebung, bestehend aus Validierung, Nutzung in kontextsensitiven Editoren, und passabler Browser-Ansicht.
- Damit kann man durchaus eine Dienstleistung für Institutionen und Autoren anbieten! Denn wenn jemand eine wissenschaftliche Karriere machen will, muss er in Journalen publizieren und auf einschlägigen Konferenzen vortragen - und braucht JATS für die Einreichung zum Preview/Review und zur Publikation
- Die SimpleJATS -DTD läßt sich nun dank verständlicher Form jederzeit auf noch fehlende Funktionen erweitern und durch ihre Kompaktheit effizient nutzen
- Sollte sich JATS im internationalen Verlagswesen als gemeinsamer Standard etablieren, wäre die Nutzung höchst sinnvoll, zum Beispiel für die Digitalisierung alter Artikel, die dann gemeinsam nutzbar sind
- Tabellen sind sehr einfach zu konstruieren, hier wird einfach der HTML-Syntax (table, tr, td) verwendet, den man in einigen Minuten beherrscht oder sowieso kennt. Darüber hinaus gehende Gestaltungsmöglichkeiten gibt es mangels CSS-Styles nicht

4.4 TEI (Text Encoding Initiative)

TEI ist ein von der gleichnamigen Organisation TEI entwickelter Dokumentstandard, der vorzugsweise in der Geisteswissenschaften verwendet wird.

http://www.tei-c.org/Guidelines/Customization/Lite/

4.5 DocBook ?

DocBook ist ein Dokumentstandard, der sich besonders zur Erstellung von Büchern, Artikeln und Dokumentationen im technischen Umfeld (Hardware oder Software) eignet. DocBook ist ein offener Standard, der von der Organization for the Advancement of Structured Information Standards (OASIS) gepflegt wird.

5 Webservices

In Arbeit!

5.1 OAI-PMH Einbindung ?

Im wesentlichen handelt es sich bei diesem „Protocol for Metadata Harvesting“ um die Übermittlung von Dublin Core Metadaten an einen sogenannten Harvester. Dazu müsste diskutiert werden, ob die existierende Version nur auf bekannte HTML-Metadaten zugreift, oder flexibel parametrierbar ist.

Der zu übermittelnde „XML-Record“ könnte grundsätzlich per XSLT-Transformation aus der Index-Datei einer Publikation erstellt werden, bzw. aus den XML-Objekten einer Publikation. Erforderlich dazu wären aber zusätzliche Metadaten, die von der Redaktion eingepflegt werden müssten, sowie statische Metadaten, die einmalig im XSLT-Stylesheet definiert werden.

> Siehe dazu
> http://www.openarchives.org/OAI/openarchivesprotocol.html
>
> OAI for Beginners - the Open Archives Forum online tutorial
> https://www.oaforum.org/tutorial/

5.2 RDF (Triplestore) Export ?

RDF Triple Store ist im wesentlichen ein Datenbank-Konzept für die Bereitstellung und Nutzung von RDF-Triples (Subject-Predicate-Object). Es ist in Zusammenhang mit „Big Data“- und semantischen Anwendungen von Bedeutung. RDF erweitert den begrifflichen Kontext um eine Aussage und ist somit für Schlussfolgerungen (Reasoning) geeignet. Überlegungen in diese Richtung sind aber verfrüht, da hier erst solide Kompetenz aufgebaut werden muss.

> Siehe dazu
> http://ontotext.com/documents/white_papers/The-Truth-About-Triplestores.pdf

6 Nützliche Links

Math ML siehe
https://www.w3.org/Math/
http://www.math-it.org/Publikationen/MathML_de.html
http://www.cs.tut.fi/~jkorpela/math/

MathJax
http://cdn.mathjax.org/
http://docs.mathjax.org/en/latest/tex.html#environments

SVG (Scalable Vector Graphics)
http://www.w3schools.com/svg/default.asp

TEI versus JATS
What JATS Users should Know about the Book Interchange Tag Suite (BITS)
http://www.ncbi.nlm.nih.gov/books/NBK159737/

Superimposing Business Rules on JATS
http://www.ncbi.nlm.nih.gov/books/NBK279902/

Easily convert Word manuscripts into JATS or TEI XML
http://www.ictect.com/JATS-XML

Papers and Presentations by Mulberry Staff
http://www.mulberrytech.com/papers/index.html