

Was wir von DITA lernen können

Fachvortrag tekomp Jahrestagung,
Stuttgart, 09.11.2016
K. Schrempp, PANTOPIX GmbH & Co. KG

Was wir von DITA lernen könnten – wenn wir denn wollten!

OVIDIUS/camp

Berlin, 13.04.2016

K. Schrempp, PANTOPIX GmbH & Co. KG



- Ausgangspunkt:
Fast ideologische Auseinandersetzung während der letzten
tekomm und in den sozialen Netzen:

DITA vs. CMS

Ziele

- Eine Auswahl von Methoden und die Idee von DITA genauer betrachten
- Nachdenken anregen, ob und wie sich das mit den ausgereiften CMS zusammenführen lässt
- Sichtbar machen, dass wir in Einzelfällen bereits mehr DITA machen, als wir denken oder zugeben wollen



- Ohne PCs
- C64, Großrechner
- PCs und Textverarbeitung
- DTP-Programme und erste CMS

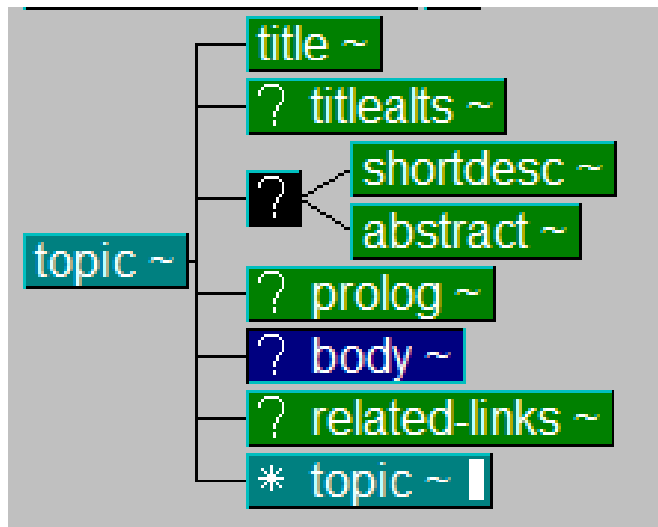
- SGML
 - Einsatz bei Luftfahrt und Militär
 - Druckerzeugnisse
 - DocBook
 - Komplexe Publikationen (DSSSL)

- Übergang zu XML
 - Erste Schritte in den 90er Jahren
 - Heute ist XML die Basis von fast allem bzw. die Antwort auf fast alles.

- Funktionsdesign®
 - Prof. Robert Schäflein Armbruster und Prof. Jürgen Muthig
 - Tübinger Linguisten
 - Anwendung der Methoden der Textanalyse für die Verständlichkeit von Texten auf Gebrauchstexte
- Problemtypologie
- Funktionale Einheiten und Sequenzmuster
- Herausforderung: Die Ebene zwischen Dokumenten und Sequenzmustern

7.9 Zielangabe (Funktionale Einheiten)

| | |
|---|---|
| Abkürzung | ZA |
| Wofür? | Benennt das Ziel einer (komplexen) Handlung |
| Wofür nicht? | -- |
| Sequenzierung | Erstes, notwendiges Element einer Handlungsanleitenden Sequenz |
| Innere Struktur und Formulierung | Substantiv und Verb |
| Kennzeichnung | Abhängig von der Kapitelebene: (Teil) Kapitelüberschrift oder Marginalie (!!!) |
| Beispiel | Gerät kalibrieren |



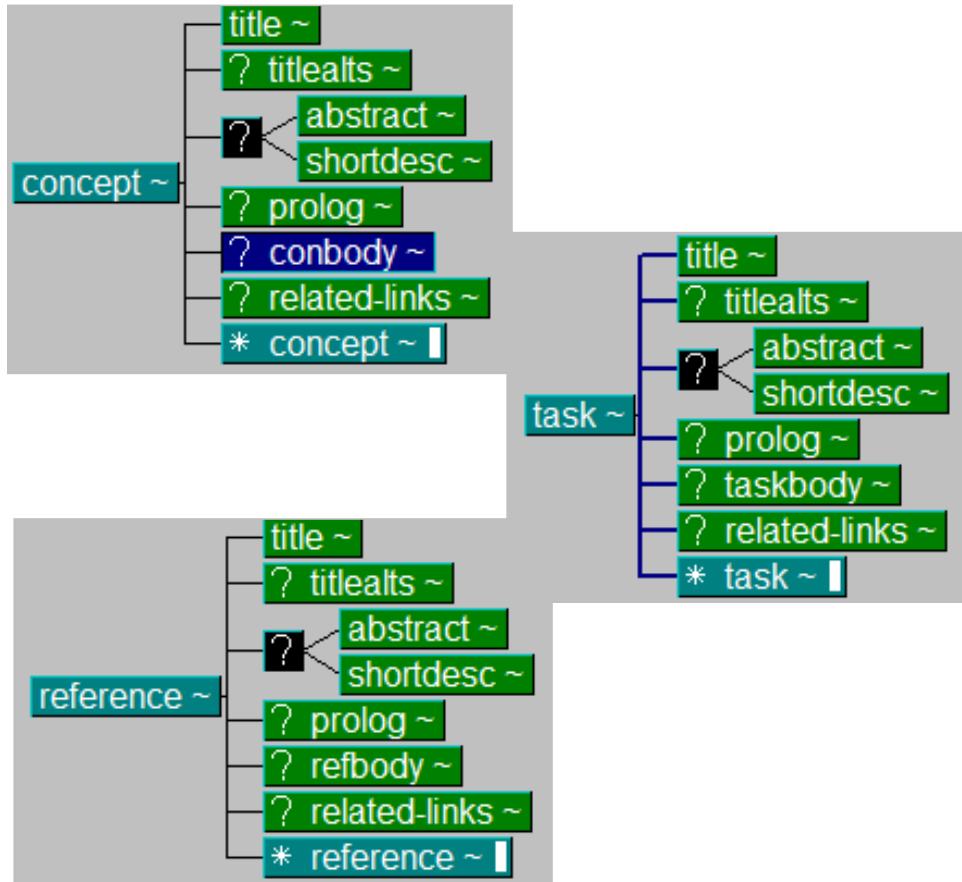
■ DITA

- Modell und Methoden, die Filesystem-basiert beschrieben werden.
- DITA kann in dieser Form auch genutzt werden.
- Wir interessieren uns für das Modell und die Methoden.

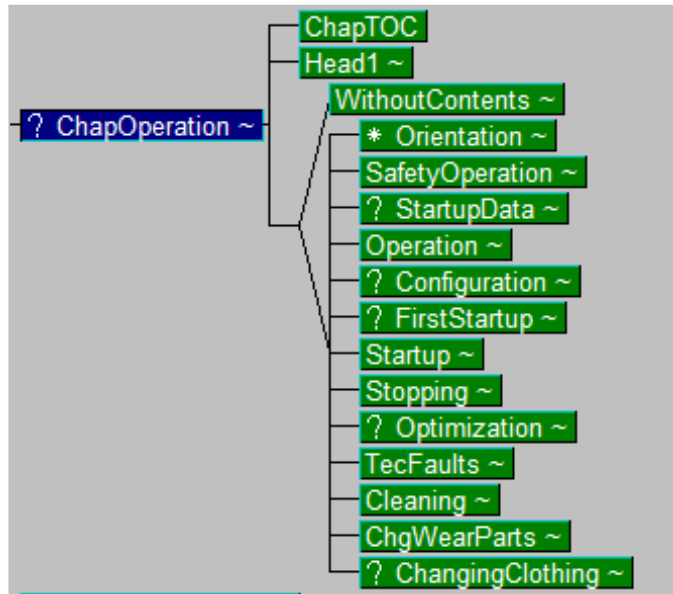
(versus)

■ CMS

- Software, die Objekte in einer Datenbank verwaltet.
- Inhalte werden im Format XML erfasst und verwaltet.
- Funktionen wirken auf das Objekt- und das XML-Modell.
- Die verschiedenen Systeme unterscheiden sich grundlegend in diesen Modellen, ihrer Verknüpfung und der Umsetzung der Funktionen.

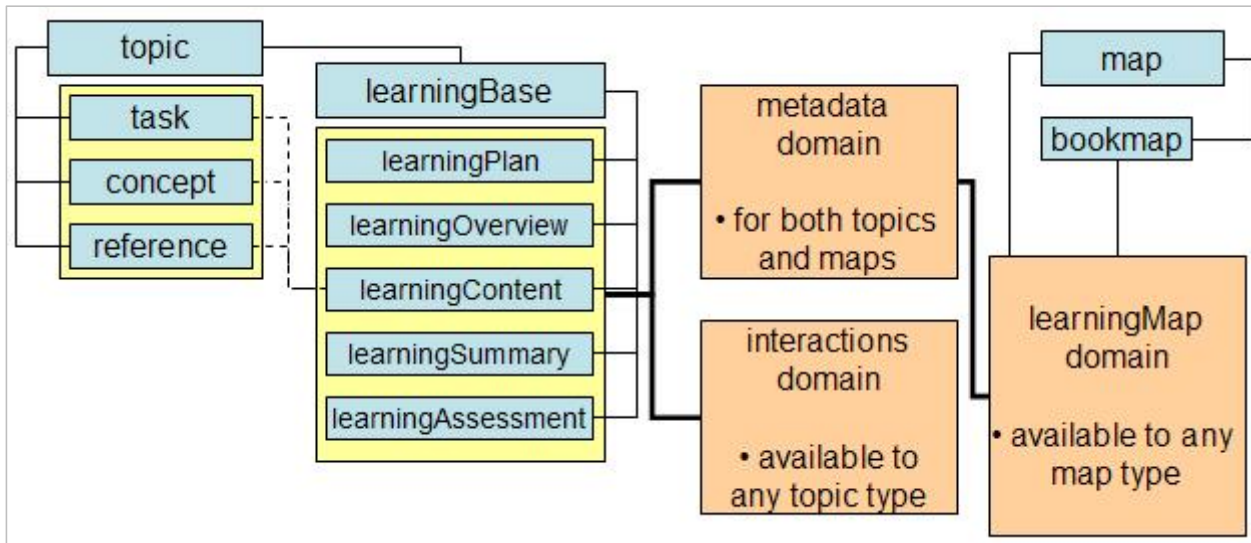


- Offenes und transparentes Datenmodell
- Topic-Orientierung
 - Die Topics folgen einem kommunikativen Bedarf, der sich in ihrer Semantik abbildet.
 - Die all-in-one Datenmodelle für Inhalte der meisten CMS tun das nicht – was nicht stört, solange wir nur Papier produzieren
 - In einer Welt interaktiver Kommunikation wird sich das als hinderlich herausstellen, da wir die Darstellung dem kommunikativen Zweck anpassen: Granularität und Textfunktion spielen eine entscheidende Rolle.
- Erweiterbarkeit
 - DITA im Rahmen des Standards mit Grenzen
 - CMS: Im Rahmen von Projekten basierend auf Enterprise-Lösungen
 - Die Zu- bzw. Wegschaltbarkeit von Domänen (lokalen oder globalen Semantiken) wäre ein hervorragendes Feature für ein CMS:
 - Software
 - Maschinenbau



- Das Funktionsdesign® hat in Hinblick auf Mobile Dokumentation nicht an Bedeutung verloren – im Gegenteil

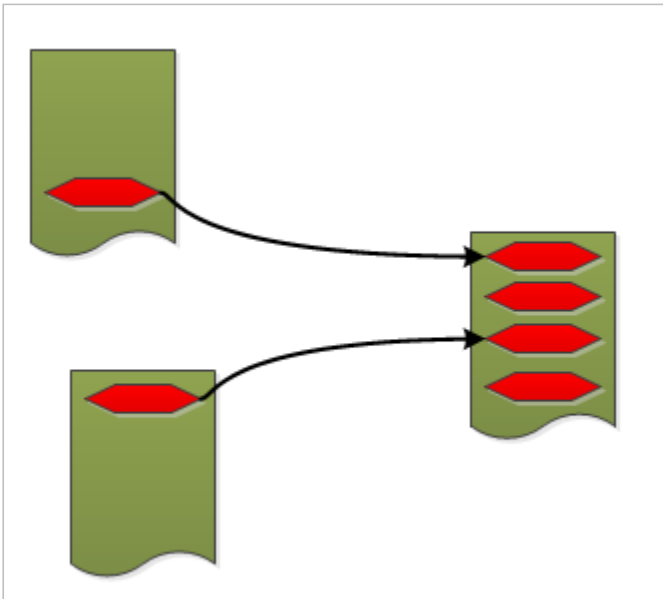
=> Individuelle (oder lokale) Semantik in Schematron-Regeln und/oder in Templates verpacken, nicht in die Datenarchitektur



- DITA hat ein sehr starkes Modell für Lerninhalte
- Semantik der CMS dort erweitern, wo es einen globalen Bedarf gibt.
 - Dabei geht es insbesondere um eine Integration der Modellerweiterungen in die bestehende Welt.
- Weitere Themen
 - Service und Diagnose
 - Case Based Reasoning
 - Entscheidungsbäume
- Standardbasierte Systeme erleichtern den Austausch und die Zusammenstellung aus verschiedenen Quellen



- Wiederverwendung
 - Variantenmanagement
 - Hierarchische Metadaten
 - Beziehungswissen zwischen Metadaten
 - Objektwelt
 - Verlinkung
 - Grenzen der Transparenz
 - DITA als Standard
-
- Fazit



- Vorwurf: Das Wiederverwendungsmodell von DITA ist nicht revisionssicher
- Die Methode „conref“ zur Wiederverwendung von Blöcken mit einer Granularität kleiner als ein Topic funktioniert im Filesystem sinnvoll (und syntaktisch korrekt) nur mit sogenannten Warehouse-Topics. Dieses Vorgehen ist in der Tat nicht revisionssicher.
 - Freigabe, Versionierung, Verwendungsnachweis sind rein Filesystem-basiert (fast) nicht abbildbar.
- Kollidiert das mit einem Einsatz eines CMS?
 - **Fragmente werden einzeln verwaltet – conrefs einzeln ablegen.**
 - Bei der Bereitstellung (Export, Publikation) die conrefs in Topics verpacken und den Link erzeugen*
 - Das CMS bietet dann eine optimierte Möglichkeiten der Verwaltung, Suche und Einbindung in alle anderen Prozesse.



Kernpunkt:

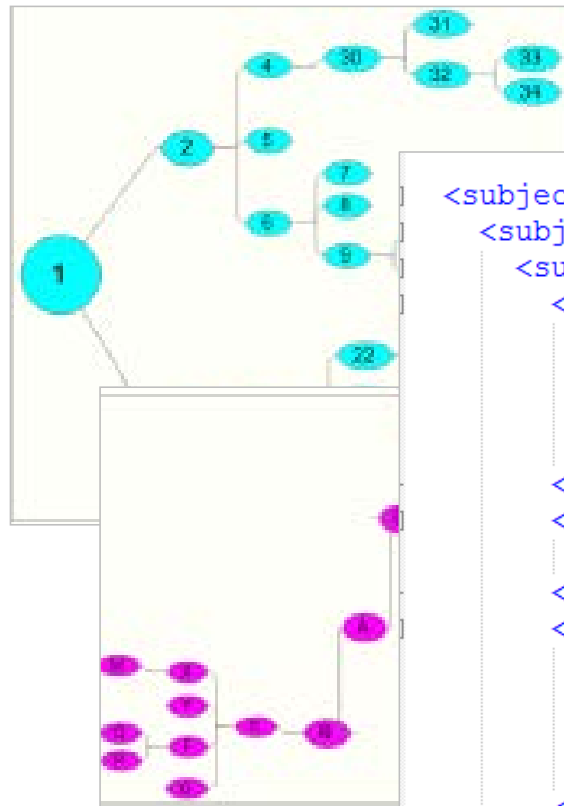
- DITA ist ein Modell.
- Wir verstoßen nicht gegen das Modell, wenn wir es geeignet abbilden und die Stärken eines CMS nutzen.
- Wir müssen uns nur darüber einigen, was wir „drinnen“ und was wir „draußen“ benötigen.
- Wir können immer das Modell publizieren

Sample:

```
<map domains="(map mapgroup-1235)">
  ...
  <keydef keys="load-toner"
    href="topics/tasks/model-1235-load-toner-proc.dita"
  />
  ...
</map>

<map domains="(map mapgroup-4545)">
  ...
  <keydef keys="load-toner"
    href="topics/tasks/model-4545-load-toner-proc.dita"
  />
  ...
</map>
```

- DITA bietet verschiedene Möglichkeiten des Variantenmanagements, die kombiniert werden können.
- Erweiterte Funktionalität der Funktion „**Variable**“
 - In den Inhalten werden Variable erfasst:
 - Die Variablen können auf folgende Objekte zeigen:
 - Techn. Werte im klassischen Sinn
 - Topics, XML-Blöcke oder Elemente
 - PDFs oder andere Medien
 - Weiter lokale oder globale Ziele
 - Die Belegung der Variablen wird pro Variante in einer Map vordefiniert, die den aktuell in Bearbeitung befindlichen Inhalten „vorangestellt“ wird. („doppelte Indirektion“)
 - Die Belegung der Variablen ist im Authoring-Prozess sichtbar.
- **Umfang der möglichen Zielobjekte von Variablen erweitern und Belegungen von Variablen im Redaktionsprozess für den Redakteur sichtbar machen.**



- DITA hat ein transparentes Modell: subjectScheme
- Taxonomien über Standardmodell austauschbar machen
 - Import und Export!

```
<subjectHead navtitle="Produktkomponente">  
  <subjectdef keys="pclass.laser" navtitle="Lasengeruert">  
    <subjectdef keys="pclass.optic-box" navtitle="Optikraum/Laserkopf">  
      <subjectdef keys="pclass.pump-unit" navtitle="Pumpeinheit">  
        <subjectdef keys="pclass.pump-module" navtitle="Pumpmodul (PM1...PMx)"/>  
        <subjectdef keys="pclass.connector" navtitle="Steckverbinder CAN, PMx"/>  
        <subjectdef keys="pclass.filter-fresh-air" navtitle="Filter für Reinluft-Austritt"/>  
        <subjectdef keys="pclass.temp-and-humidity-B5" navtitle="Temperatur- und Feuchtesensor (B5)"/>  
      </subjectdef>  
    <subjectdef keys="pclass.water-distributor" navtitle="Wasserverteiler">  
      <subjectdef keys="pclass.water-distributor.all" navtitle="alle"></subjectdef>  
    </subjectdef>  
    <subjectdef keys="pclass.beam-source" navtitle="Strahlquelle">  
      <subjectdef keys="pclass.seedlaser" navtitle="Seedlaser"/>  
      <subjectdef keys="pclass.faser" navtitle="Faser"/>  
      <subjectdef keys="pclass.laser_module" navtitle="Lasier-Modul"/>  
      <subjectdef keys="pclass.cavity" navtitle="Kavität"/>  
    </subjectdef>  
  </subjectdef>  
</subjectdef>
```

```
<subjectRelTable>
  <subjectRelHeader>
    <subjectRole>
      <subjectdef keyref="models">
        <hasRelated keyref="contains-component">
          <subjectdef keyref="product-components" />
        </hasRelated>
      </subjectdef>
    </subjectRole>
    <subjectRole>
      <subjectdef keyref="product-components" />
    </subjectRole>
  </subjectRelHeader>
  <subjectRel>
    <subjectRole>
      <subjectdef keyref="taifun" />
    </subjectRole>
    <subjectRole>
      <subjectdef keyref="engine-compartment">
        <subjectdef keyref="engine-compartment-bonnet" />
        <subjectdef keyref="engine-oil" />
        <subjectdef keyref="brake-fluid" />
        <subjectdef keyref="fuse" />
      </subjectdef>
    </subjectRole>
  </subjectRel>
</subjectRelTable>
```

- subjectSchemes können Beziehungswissen unter Metadaten abbilden!
- Applikationen müssen noch lernen, dieses Wissen zu interpretieren.
- iiRDS kann dieses Wissen transportieren
- CMS mit geeigneten Editoren bzw. mit einer passenden Visualisierung versehen.

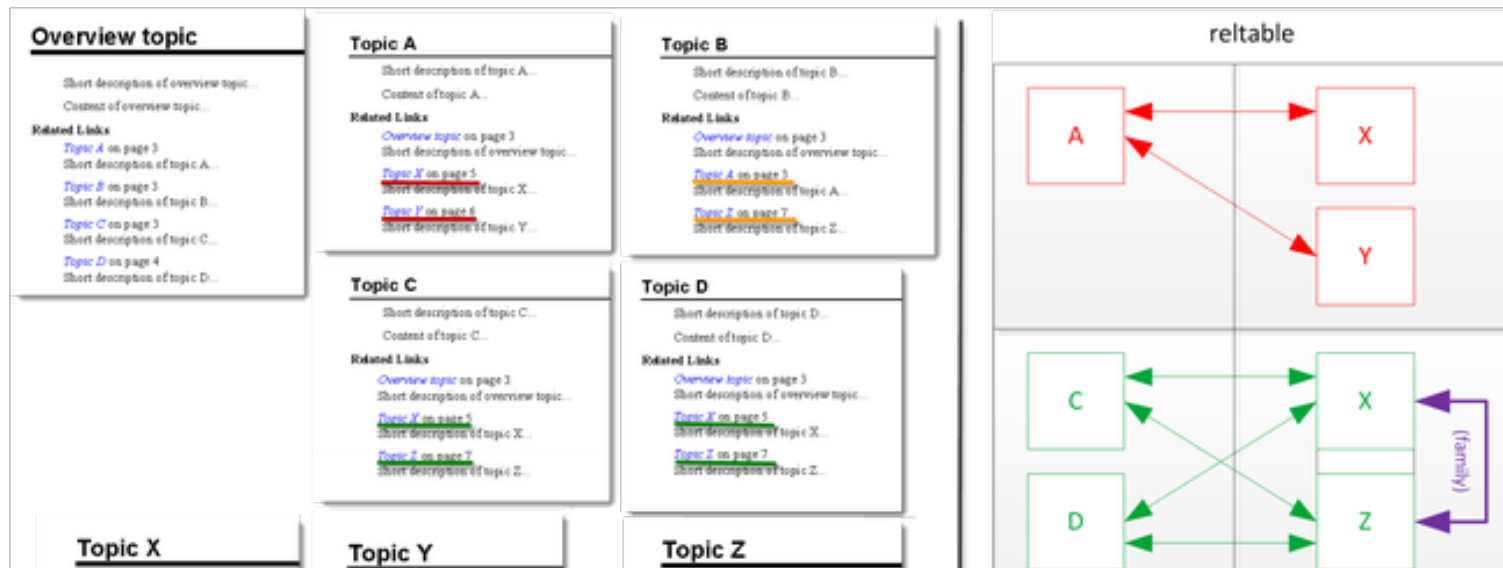
- Ziel: allgemein: Links so weit wie möglich aus Inhalten herauslösen oder zumindest verallgemeinern:
 - Parametrierbare Links (die noch im Inhalt stehen)
 - Erzeugen von Links über Metadatenähnlichkeit

DITA Lösung 1:

- Block <related-links>

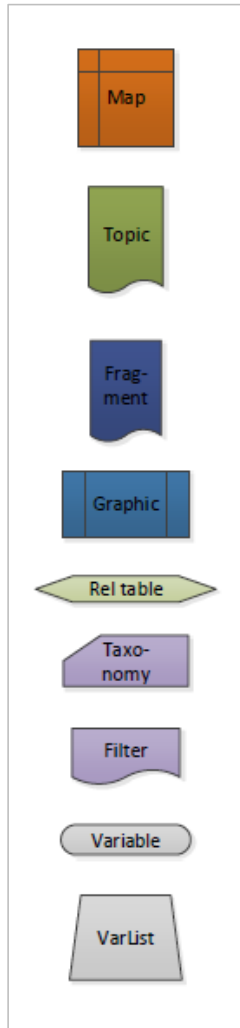
DITA Lösung 2 (besser):

- Links werden als retables über die Inhalte gelegt
- Nutzung des Variantenmanagement (siehe oben)

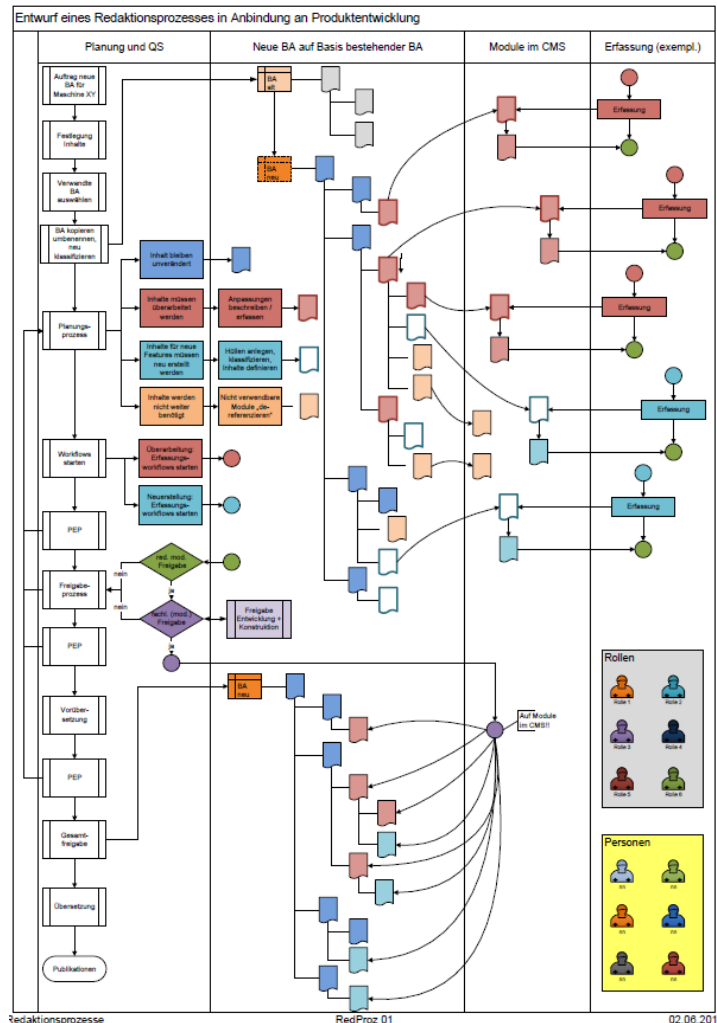


- CMS mit geeigneten Editoren bzw. und passenden Visualisierung versehen.

Source: think-dita.com, 06.11.2016



- Da in DITA alle Informationen im Dateisystem liegen, sind sie per se als einzelne Objekte zugänglich und können editiert, freigegeben, versioniert und (wo sinnvoll) übersetzt werden.
- **Mindestens folgende Objekte zu Objekten im CMS machen:**
 - Filter
 - Variablenlisten
 - Taxonomiedefinitionen
 - Linklisten/-tabellen
- Damit können diese Objekte folgenden Prozesse unterworfen werden:
 - Freigabe und Versionierung
 - Übersetzung (wenn sinnvoll) mit Freigaben
 - Wiederverwendung
 - z. B. einzelner Filter für bestimmte Kriterien in Kombination



- Für die Prozesse selber (und nicht nur für die Objekte der vorigen Folie) ist der Einsatz eines CMS nahezu unabdingbar.
 - Oberfläche
 - Funktionen
- In kollaborativen Umgebungen sollten wir die Funktionen und die Architektur eines CMS nutzen können, um die Nutzer effizient und nachhaltig zu unterstützen.
- D. h. es interessiert aus meiner Sicht nicht, ob die Funktion formal wie im Dateisystem abgebildet wird, sondern dass sie über die Architektur und das Datenmodell des CMS nutzbar ist.
 - Ist das dann noch DITA?

(Wiederholung)

- DITA ist ein Modell.
 - Wir verstoßen nicht gegen das Modell, wenn wir es geeignet abbilden und die Stärken eines CMS nutzen.
 - Wir müssen uns nur darüber einigen, was wir „drinnen“ = im **CMS** und was wir „draußen“ = zur **Weitergabe** und zum **Austausch** benötigen.
 - Wir könn(t)en immer das Modell als Output publizieren.
- => Trennung von Authoring und Publikations-/Renderingprozess
- Wir haben einen Vorteil, wenn wir ein gemeinsames Modell und seine Funktionen nutzen, ausgeprägt durch die individuellen Architekturen eines CMS:
 - Vorgehen und Aufwände für Konfiguration
 - Verfügbare Tools und transparente (Publikations-) Prozesse
 - Austausch und Weitergabe von publizierten Daten an Drittsysteme funktioniert nur/besser auf der Basis eines Standards





- Wenn wir Äpfel mit Birnen vergleichen, gewinnen wir keine Erkenntnis.
- Wenn wir ein etabliertes Modell richtig verstehen, können wir es gezielt und mit Vorteil umsetzen.
- Die methodisch-funktionalen Aspekte von DITA sind unumgänglich, wenn wir mit hoher Flexibilität Informationen erstellen und verteilen wollen.
- Damit wird die Komplexität von DITA zu einer scheinbaren Komplexität, der wir uns in jedem Fall stellen müssen.
- Weitergabe von publizierten Daten an Drittsysteme funktioniert nur/besser auf der Basis eines Standards.
- Austausch von Inhalten mit Drittsysteme funktioniert nur auf der Basis eines Standards.
- Intelligente Information ist unabhängig vom Format, aber ein gemeinsames Format erleichtert die Arbeit.

**Lassen Sie uns künftig konstruktiver hinschauen und diskutieren –
DITA modelliert Aspekte, die wir benötigen!**

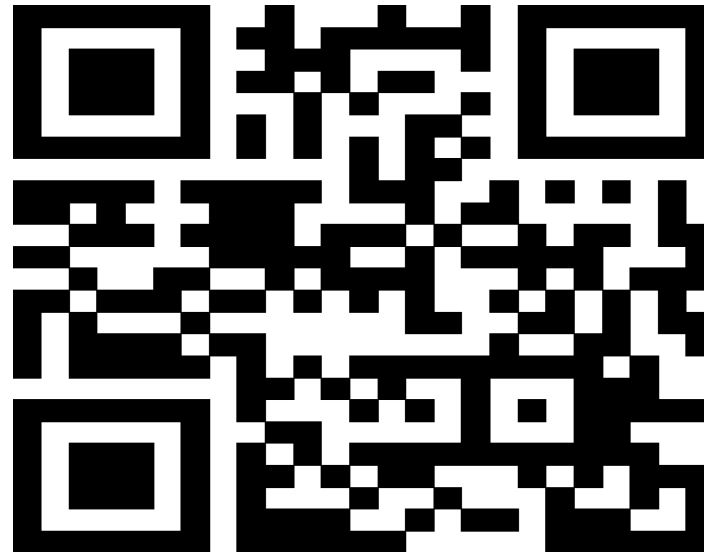
PANTOPIX GmbH & Co. KG
Karsten Schrempp
Mobil: +49 170 386 60 91
karsten.schrempp@pantopix.de



Ihre Meinung ist uns wichtig! Sagen Sie uns bitte, wie Ihnen der Vortrag gefallen hat. Wir freuen uns auf Ihr Feedback per Smartphone oder Tablet unter

<http://in13.honestly.de>

oder scannen Sie den QR-Code



Das Bewertungstool steht Ihnen auch noch nach der Tagung zur Verfügung!