

Blogpost - Gendern smart gelöst

Contents

Gendern smart gelöst: Eine technische Lösung für PDF-Publikation und Content Delivery im Web.....	3
--	----------

Gendern smart gelöst: Eine technische Lösung für PDF-Publikation und Content Delivery im Web

Wie Sie Gendern bei PDF-Publikation und Content Delivery im Web technisch angehen können
In diesem Blogbeitrag stellen wir eine technische Lösung für das Gendern im Web und beim Publizieren von PDF vor.



In seiner [Ausgabe 2021/10](#) titelt der Spiegel “Gendergerechte Sprache - Ist das noch Deutsch? Der Kulturkampf um die Sternchen”. Wir von c-rex.net haben uns gefragt, wie unsere Antwort auf diesen Kulturkampf und den gesellschaftlichen Diskurs aussehen könnte und präsentieren hier unsere Antwort: Gendern bei der Publikation nach PDF und beim Content Delivery im Web. Eine technische Lösung.

Spoiler: Gleichzeitig mit der Veröffentlichung dieses Blogbeitrags haben wir die Lösung, über die wir berichten, in unserem Blog umgesetzt. Meine Kollegin hat diesen Blogbeitrag (sowie rückwirkend auch alle bisherigen Blogbeiträge) gendergerecht geschrieben. In jedem Beitrag können Sie einstellen, in welcher Variante Sie diesen Text lesen möchten.

Worum es beim Gendern geht

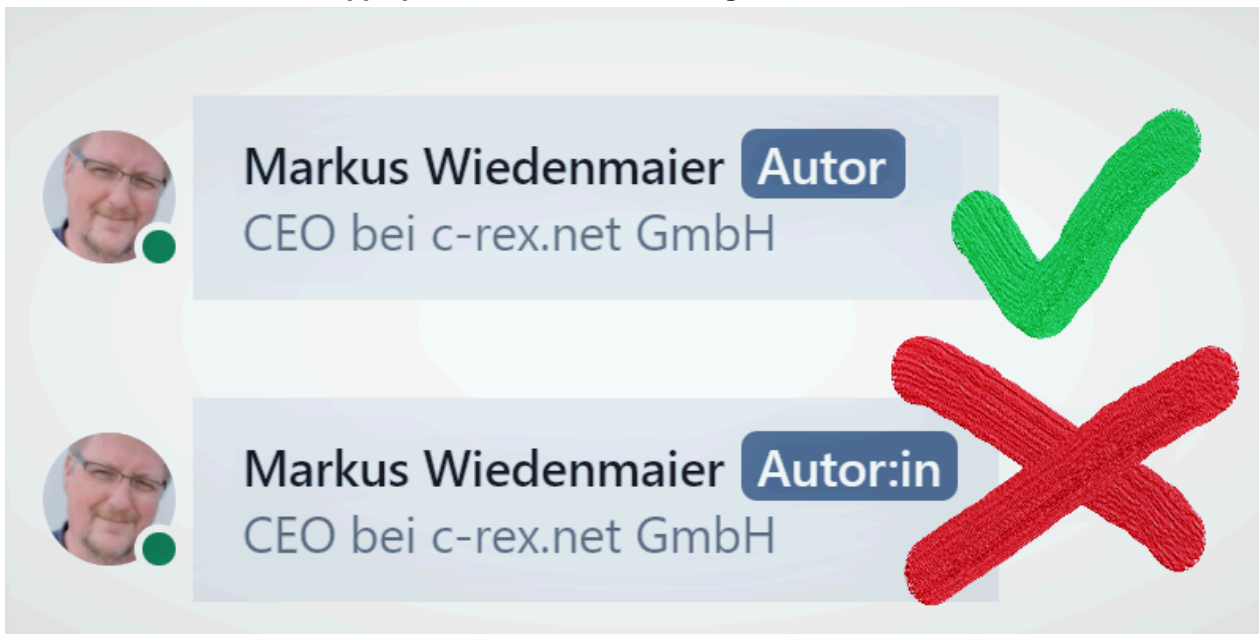
"gender" ist ein englischer Begriff für "Geschlecht", wobei hier das gelebte Geschlecht gemeint ist im Gegensatz zu "sex", welches sich auf die Geschlechtsmerkmale bezieht. Beim "Gendern" geht es in den Diskussionen darum, Personen nicht mehr mit dem generischen Maskulinum "mitzumeinen", sondern einen Weg zu finden, um in der deutschen Sprache alle, unabhängig ihres (gelebten) Geschlechts, anzusprechen.

Hierzu wird dem Maskulinum in verschiedenen Formen die weibliche Endung angehängt (z.B. AutorIn, Autor*in, Autor:in). Es gibt auch heiß diskutierte Vorstöße, ein generisches Femininum einzuführen. So geschehen im Oktober 2020, als ein [Gesetzentwurf des Justizministeriums](#) in ausschließlich weiblicher Form geschrieben wurde.

Natürlich ist es schön, nett und in vielen Fällen sogar wichtig, jeden so persönlich wie möglich anzusprechen, um niemanden wegen seines Geschlechts mit der Sprache auszugrenzen. Das ist besonders schwierig, wenn man die angesprochenen Personen nicht kennt; und deshalb wird die Diskussion, wenn auch manchmal etwas übertrieben, zu Recht geführt.

Bei Audi heißt [die neue Richtlinie](#) mit dem Apell an Mitarbeiter, bei der Kommunikation gendergerechte Sprache zu verwenden: "Vorsprung beginnt im Kopf!" Ein 13-seitiges Dokument, das in einer **einjährigen(!) Arbeitsgruppe** entwickelt wurde. Aber der Slogan von Audi ist doch "Vorsprung durch Technik"! Wenn man ein Jahr lang in einer Taskforce an einem Konzept arbeitet, warum nimmt man dann nicht das Credo des Unternehmens und setzt das Konzept gemäß diesem Leitsatz auch entsprechend technisch um (oder lässt es umsetzen)?

Gender-Sternchen oder Doppelpunkte nur da, wo es nötig ist



Wenn mich jemand kennt, möchte ich als was oder wie ich bin angesprochen werden, und nicht gegendert oder in sonstiger Form. Ich bin Autor oder Entwickler oder eben Autorin oder Entwicklerin, aber nicht Autor:in oder Entwickler:in. Werde ich also persönlich angeschrieben oder erwähnt, möchte ich auch in der zu mir passenden Form angesprochen werden. Das beginnt schon beim Brief, der oft so beginnt "sehr geehrte/r Frau/Herr Wiedenmaier" oder wie hier auf LinkedIn

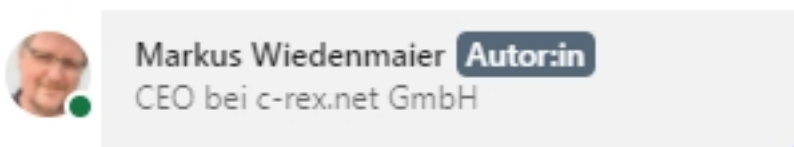


Abbildung 1: Gendern auf LinkedIn

Da frage ich mich dann manchmal, liegt es an den Systemen, also sind Entwickler des jeweiligen Portals dafür verantwortlich, oder möchte man hier einfach nur zeigen, dass man auch das Gender-Sternchen bzw. den Gender-Doppelpunkt beherrscht. Ich bin eben Autor und nicht Autor:in.

Da eine gendergerechte Ansprache vielen Menschen wichtig ist, haben wir uns ebenfalls dazu entschieden, in unserer Kommunikation die gendergerechte Sprache zu nutzen, und in unserer Dokumentation sowie in unserem deutschen Webauftritt gendergerecht zu kommunizieren. Die Entscheidung, wie das Ergebnis dargestellt werden soll, überlassen wir dabei unserer Zielgruppe und lassen Leser den Text in der Variante lesen, die ihnen am Besten gefällt. Wir gehen damit über den Ansatz bei LinkedIn oder sonstigen Portalen weit hinaus. **Schauen wir uns an, wie es technisch funktioniert.**

Eine Lösung für Systeme mit Benutzerprofilen: vorhandene Informationen auswerten

Gerade im Web sind wir den Systemen mittlerweile bekannt. Je mehr wir uns im Netz bewegen, umso transparenter werden wir. Systeme wissen, nach was ich vor kurzem gesucht habe, was ich gern esse, wie mein sozialer Status ist. Auf Basis dieser Daten wird mir z.B. regelmäßig entsprechende zu mir passende Werbung eingeblendet, wo ich mich manchmal frage: "Woher wissen die das schon wieder?!"

Wenn mich also Systeme kennen, wissen sie auch, welches Geschlecht ich habe oder lebe. Im einfachsten Fall, indem ich das in meinem Benutzerprofil hinterlegt habe. Wenn ein System das also weiß, wieso werde ich, wie auf LinkedIn als Autor:in und nicht als Autor getagged? In diesem Fall wäre das noch ziemlich einfach, weil dieses Tag im System hinterlegt ist und problemlos mit meinem Profil verknüpft werden kann, um eben den für mich richtigen Begriff anzuzeigen.

Ein Lösungsansatz für redaktionelle Texte: Metadaten verwenden und bei der Contentgenerierung auswerten

Aber wie sieht es nun mit Freitexten z.B. in Blogposts, in Webartikeln oder gar in der technischen Dokumentation aus? Hierfür gibt es Technologien und Standards, mit denen Lösungsansätze für das Problem gefunden werden können. Nicht ganz überraschend heißt eine dieser Lösungen [XML](#). Eine weitere Lösung ist das Terminologiemanagement in Form des [TBX](#) (TermBase eXchange) ISO-30042-Standards, ebenfalls ein XML-Standard.

Versehen wir doch die gender-spezifischen Begriffe mit einem Metadatum und lassen das jeweilige Publikationssystem entweder bei der Generierung oder gerade im Web on-demand entscheiden, welcher gender-gerechte Begriff der passende ist und angezeigt werden soll.

Geschlechterspezifische Auszeichnung von Begriffen mit TermBase eXchange

In der XCS-Datei (extensible constraint specification), die eine Erweiterung zur *TBxcoreStructV02.dtd* darstellt, sind bereits Typen zur geschlechterspezifischen Auszeichnung von Begriffen vorgesehen. Und genau diese können wir im Rahmen des Terminologiemanagements verwenden:

```
<termNoteSpec name="grammaticalGender" datcatId="ISO12620A-020202">
  <contents datatype="picklist" forTermComp="yes">masculine feminine
  neuter otherGender</contents>
</termNoteSpec>
<termNoteSpec name="grammaticalNumber" datcatId="ISO12620A-020203">
  <contents datatype="picklist" forTermComp="yes">singular plural dual
  mass otherNumber</contents>
</termNoteSpec>
<termNoteSpec name="termType" datcatId="ISO12620A-0201">
  <contents datatype="picklist">abbreviation acronym clippedTerm
  commonName entryTerm
  equation formula fullForm initialism internationalScientificTerm
  internationalism
  logicalExpression phraseologicalUnit partNumber shortForm shortcut
  sku symbol
  synonym synonymousPhrase standardText string transliteratedForm
  transcribedForm
  variant </contents>
</termNoteSpec>
```

Bereits im TBX-Standard können wir also die Metadaten `grammaticalGender` (männlich, weiblich) und `grammaticalNumber` (Einzahl/Mehrzahl) nutzen, um Begriffe (Terme) entsprechend auszuzeichnen. Ebenso können wir den `termType` (`shortForm`, `fullForm`) verwenden, um eine abgekürzte Form (Sternchen oder Doppelpunkt) oder die ausgeschriebene Variante zu definieren. Eine beispielhafte Umsetzung mit dem Term *Autor* könnte so aussehen:

```
<termEntry id="term_g_author">
  <descrip type="definition">Der Autor ist der Schreiber eines Textes</
  descrip>
  <langSet xml:lang="de-DE">
    <tig>
      <term>Autor</term>
      <termNote type="grammaticalGender">masculine</termNote>
      <termNote type="grammaticalNumber">singular</termNote>
    </tig>
    <tig>
      <term>Autorin</term>
      <termNote type="grammaticalGender">feminine</termNote>
      <termNote type="grammaticalNumber">singular</termNote>
    </tig>
    <tig>
      <term>Autor:in</term>
      <termNote type="grammaticalGender">neuter</termNote>
      <termNote type="grammaticalNumber">singular</termNote>
      <termNote type="termType">shortForm</termNote>
    </tig>
    <tig>
      <term>Autor oder Autorin</term>
      <termNote type="grammaticalGender">neuter</termNote>
      <termNote type="grammaticalNumber">singular</termNote>
      <termNote type="termType">fullForm</termNote>
    </tig>
    <tig>
      <term>Autoren</term>
      <termNote type="grammaticalGender">masculine</termNote>
      <termNote type="grammaticalNumber">plural</termNote>
    </tig>
  </langSet>
</termEntry>
```

```

</tig>
<tig>
  <term>Autorinnen</term>
  <termNote type="grammaticalGender">feminine</termNote>
  <termNote type="grammaticalNumber">plural</termNote>
</tig>
<tig>
  <term>Autor:innen</term>
  <termNote type="grammaticalGender">neuter</termNote>
  <termNote type="grammaticalNumber">plural</termNote>
  <termNote type="termType">shortForm</termNote>
</tig>
<tig>
  <term>Autoren oder Autorinnen</term>
  <termNote type="grammaticalGender">neuter</termNote>
  <termNote type="grammaticalNumber">plural</termNote>
  <termNote type="termType">fullForm</termNote>
</tig>
</langSet>
</termEntry>

```

Mit diesem Termeintrag haben wir Varianten für den Begriff *Autor* definiert und ihm entsprechende Bedeutungen in Form von Meta-Informationen zugewiesen. Somit haben wir quasi eine Datenbank geschaffen, mit der ein – wie auch immer geartetes – System die entsprechenden Gender-Variationen erkennen kann.

Ein klassisches Terminologieprüfungssystem hilft uns aber in diesem Usecase nicht weiter, da es uns in der Regel nur sagt, welchen dieser Begriffe wir in der Redaktion, also beim Schreiben verwenden sollten, nicht aber, welcher in einer personalisierten Publikation verwendet werden soll.

Eine Terminologieprüfung hilft uns, im Erstellungsprozess sicherzustellen, dass wir z.B. die gender-neutrale Version verwenden, wenn wir schreiben. Damit ist unser Inhalt aber eben gerade nicht personalisierbar, sondern allgemeingültig – so wie es eben definiert ist. Wir brauchen also mehr, wenn wir diese Informationen dynamisch, in einem personalisierten Kontext austauschen möchten.

Geschlechterspezifische Auszeichnung von Begriffen in xml-basierten Content-Erstellungssystemen am Beispiel DITA

Nehmen wir den einfachen Text "Wie werde ich Autor?", um das Thema in XML zu betrachten. In [DITA-XML](#) verwenden wir für die Erstellung von Absätzen das Element `<p>` und schreiben diesen Satz wie folgt:

```
<p>Wie werde ich Autor?</p>
```

Ohne weitere Logik wissen wir nicht, dass wir uns um den Begriff *Autor* in der Publikation speziell kümmern müssen. Deshalb versehen wir das Wort *Autor* mit einem entsprechenden Tag, in DITA-XML mit dem Element `<term>`, `<keyword>` oder schlicht `<ph>`.

```
<p>Wie werde ich <ph>Autor</ph>?</p>
```

Jetzt müssen wir noch eine Beziehung zu unserem Begriff in der Termdatenbank herstellen, was wir in DITA über das Attribut `@keyref` tun können. Wir verwenden also einen der Wiederverwendungsmechanismen von DITA.

```
<p>Wie werde ich <ph keyref="term_g_author">Autor</term>?</p>
```

Damit haben wir die Verknüpfung zwischen unserem Inhalt und der Terminologiedatenbank geschaffen. Noch nicht ganz! DITA versteht den TBX-Standard nicht und erwartet stattdessen eine sogenannte Key-Map, welche als Datenbasis für die über `keyref` adressierten IDs gilt. Aber wir haben ja [XSLT](#) als Standard-Technologie, über die wir aus der TBX-Datenbank unsere Key-Maps automatisch generieren können. Richtig gelesen! Wir benötigen mehrere Key-Maps, eine für jede Gender-Variante.







In unserem Fall des Beispielsatzes "Wie werde ich Autor?" sind es vier:

- die Variante für das generische Maskulinum (Autor)
- die Variante für das generische Femininum (Autorin)
- die Variante für die gegenderte Kurzform (Autor:in)
- sowie die Variante für die gegenderte Langform (Autor oder Autorin)

Jede Key-Map-Variante enthält sowohl die Form im Singular als auch im Plural und wird in der DITA-Map, in der die "Dokumentation" zusammengestellt und in Hierarchie gebracht wird, referenziert.

Gender key map - neuter-short

▼ **Audience:**

-  `term_g_author_singular` = ▶ Autor:in ◀
-  `term_g_author_plural` = ▶ Autor:innen ◀
-  `term_g_si_blogger_singular` = ▶ Smart-Information-Blogger:in ◀
-  `term_g_si_blogger_plural` = ▶ Smart-Information-Blogger:innen ◀
-  `term_g_customer_singular` = ▶ Kund:in ◀
-  `term_g_customer_plural` = ▶ Kund:innen ◀

Damit können wir diese Begriff jetzt einfach anhand ihrer ID referenzieren, und können diese auch problemlos in anderen Sprachen durch die jeweilige gültige Key-Map austauschen. **So funktioniert dann**

auch der Prozess in Übersetzungen, vor allem dann, wenn in einer Sprache nicht gegendert wird – und das dürften so ziemlich alle außer Deutsch sein.

Und was machen wir mit unserer PDF-Produktion?

Bei der PDF-Produktion erstellen wir mehr oder weniger statischen Inhalt, der teilweise auch noch gedruckt wird. Deshalb müssen und werden wir uns wegen hoher Druckkosten meist für eine der Varianten entscheiden. Bieten wir die Dokumentation zum Download im Web an, können wir aber auch Anwender entscheiden lassen und alle Varianten zum Download anbieten. Wir müssen nur wissen und entscheiden, was wir wollen.

Um das zu lösen, nutzen wir die Mechanismen des Variantenmanagements und zeichnen die Verweise auf die Key-Maps mit den gegenderten Begriffen entsprechend aus. Dazu nutzen wir in DITA das Attribut `@audience` um die Varianten entsprechend zu markieren.

```
<map>
  <title>c-rex.net Key Maps</title>
  <topicref audience="generic-masculine" href="c-rex-net-terminology_keys_gm.ditamap" format="ditamap"/>
  <topicref audience="generic-feminine" href="c-rex-net-terminology_keys_gf.ditamap" format="ditamap"/>
  <topicref audience="neuter-full" href="c-rex-net-terminology_keys_nf.ditamap" format="ditamap"/>
  <topicref audience="neuter-short" href="c-rex-net-terminology_keys_ns.ditamap" format="ditamap"/>
</map>
```

Publizieren wir nun unsere Inhalte, geben wir eine DITaval (Profiling Mechanismus in DITA) als Parameter an, um – wie in diesem Fall – die Variante mit dem generischen Maskulinum zu publizieren und alle anderen Varianten herauszufiltern.

```
<val>
  <prop action="exclude" att="audience" val="generic-feminine"/>
  <prop action="exclude" att="audience" val="neuter-full"/>
  <prop action="exclude" att="audience" val="neuter-short"/>
</val>
```

Die vier publizierten PDF-Varianten sind hier zu finden:

- [generisches Maskulinum](#)
- [generisches Femininum](#)
- [gender-neutral \(Kurzform\)](#)
- [gender-neutral \(Langform\)](#)

Und wie sieht's aus in Webportalen & Co? – Gendern im Content Delivery

Im vorherigen Abschnitt haben wir statische Publikationen wie eine PDF-Ausgabe betrachtet. Dabei müssen wir zum Generierungszeitpunkt anhand der Zielgruppe entscheiden, welche Variante des Terms wir verwenden wollen. Entweder wir schreiben etwa einen Brief an eine konkrete Person oder wir verwenden den jeweiligen gender-neutralen Begriff in einem generischen Kontext. Im Content Delivery oder generell in einem Webportal aber kennen wir Anwender erst, wenn sie am System anmelden sind. Idealerweise haben wir zudem mehrere Anwender, die online auf unsere Inhalte zugreifen möchten. Zudem sollten diese nach Möglichkeit auch entscheiden können, in welcher Variante sie den Text lesen möchten, in dem nicht sie selbst, sondern andere Gruppen angesprochen werden. Deshalb müssen wir im HTML sämtliche Varianten bereitstellen, die in diesem Kontext in Frage kommen, um sie zur Laufzeit korrekt zur Anzeige zu bringen.

```
<div class="gender-setting-generic-masculine">
  <p class="p">Wie werde ich
    <span class="term-gender generic-masculine">Autor</span>
    <span class="term-gender generic-feminine">Autorin</span>
    <span class="term-gender neuter-short">Autor:in</span>
    <span class="term-gender neuter-full">Autor oder Autorin</span>
  ?</p>
</div>
```

Diese Inhalte werden dann abhängig vom Benutzerprofil über CSS ein- oder ausgeblendet.

```
.term-gender
{border-bottom: 2px dotted red;}
.term-gender.generic-masculine
{display:inline;}
.term-gender.generic-feminine,
.term-gender.neuter-short,
.term-gender.neuter-full
{display:none;}
.gender-setting-generic-masculine .term-gender,
.gender-setting-generic-feminine .term-gender,
.gender-setting-neuter-short .term-gender,
.gender-setting-neuter-full .term-gender
{display:none !IMPORTANT;}
.gender-setting-generic-masculine .term-gender.generic-masculine,
.gender-setting-generic-feminine .term-gender.generic-feminine,
.gender-setting-neuter-short .term-gender.neuter-short,
.gender-setting-neuter-full .term-gender.neuter-full
{display:inline !IMPORTANT;}
```

In diesem Fall setzt das System im HTML einen Kontext in Form einer Klasse, die das Geschlecht der angemeldeten Anwender definiert und welche Präferenz für die Gendervariante gewählt wurde. Abhängig von dieser Einstellung werden die passenden Begriffe entsprechend ein- oder ausgeblendet.

Die Entscheidung können Anwender bei uns im Blog über ein Dropdown vornehmen. Dieses ist nur dann sichtbar, wenn der Inhalt auch tatsächlich Gendervarianten enthält:

Darstellung des Artikels:

generisches Maskulinum generisches Femininum Gender-Doppelpunkt Genderneutral

Abhängig von der getroffenen Wahl werden die Inhalte wie gewünscht dargestellt (die genderbaren Begriffe entsprechend unterstrichen markiert):

Blitzlicht*: Smarter Taschenrechner für Expertinnen

Dieser Blogbeitrag stellt eine smarte Anwendung für Heizungstechnikerinnen vor.

Aus der Kombination von isolierten Informationen, mit denen Heizungstechnikerinnen Druckverlustberechnungen für Wärmezähler durchführen, entstand eine smarte Anwendung, die Technikerinnen das Selbstausrechnen abnimmt. So geht Smart-Information im Kleinen und Feinen!

[Lesen Sie Details im Nobit-Blog](#).

*In unserer **Blitzlicht-Reihe** zeigen Smart-Information-Bloggerinnen in unregelmäßigen Abständen, kurz und knackig, was **Smart-Information** heute schon in der Praxis heißt.

Abbildung 2: Darstellung im generischen Femininum

Blitzlicht*: Smarter Taschenrechner für Expert:innen

Dieser Blogbeitrag stellt eine smarte Anwendung für Heizungstechniker:innen vor.

Aus der Kombination von isolierten Informationen, mit denen Heizungstechniker:innen Druckverlustberechnungen für Wärmezähler durchführen, entstand eine smarte Anwendung, die Techniker:innen das Selbstausrechnen abnimmt. So geht Smart-Information im Kleinen und Feinen!

[Lesen Sie Details im Nobit-Blog](#).

*In unserer **Blitzlicht-Reihe** zeigen Smart-Information-Blogger:innen in unregelmäßigen Abständen, kurz und knackig, was **Smart-Information** heute schon in der Praxis heißt.

Abbildung 3: Darstellung in einer der beiden gender-neutralen Varianten

Was machen wir mit Social Media?

Bei einem Tweet auf [Twitter](#), einem Post auf [LinkedIn](#) oder [Facebook](#) wird es naturgemäß schwierig, den Inhalt in mehreren Varianten abzusetzen. Das kann man natürlich tun, ob das aber zielführend ist, scheint doch sehr fraglich. Deshalb wird man sich hier für eine Variante entscheiden. Im Fall des c-rex.net-Bot, der in Zukunft automatisch Tweets für unseren Firmenaccount absetzen wird, wenn neue Beiträge online gehen, werden wir die Texte aus dem Teaser verwenden und dort die gender-neutrale Variante in Kurzform und mit Doppelpunkt verwenden. So können auch wir dann zeigen, dass wir den Gender-Doppelpunkt für die mit unseren Werkzeugen genierten Inhalte beherrschen.

Autorenunterstützung und Validierung

Beim Erstellen von Inhalten liegt der Fokus meist auf dem Schreiben. Und natürlich wird dabei oft vergessen, Gender-Terme entsprechend auszuzeichnen oder zu pflegen. Es braucht also einen Validierungsmechanismus, der nicht nur prüft, ob ein Begriff korrekt verwendet wird (das machen Terminologieprüfsysteme), sondern vor allem auch, ob der korrekt verwendete Begriff auch getagged wurde. Wir sind also nun vom spezifischen "Gendern" zu einer allgemeineren Frage der Terminologieprüfung gekommen. Und gerade beim Tagging haben diese Systeme ihre Schwächen, weil sie sich auf Text und nicht auf Struktur konzentrieren.

c-rex.net DTS kann Terme auf Basis einer Termdatenbank, wie sie hier erstellt wurde, taggen. Bevor ein Blog freigegeben wird, wird bei uns der Inhalt mit dieser Technologie geprüft und Begriffe, die in der Datenbank definiert aber nicht getagged sind, entsprechend durch ein **keyref**, also einen Verweis auf unsere DITA-Map, gesetzt. Zudem wird ein Eintrag in das Validierungsprotokoll geschrieben. Dieses Protokoll kann abschließend in der Redaktion nochmals auf Richtigkeit geprüft und ggf. auch der Schreibstil und die Grammatik des Satzes, in dem der Begriff verwendet wird, angepasst werden.

Deutsche Sprache, schwere Sprache oder Der Versuch eines Fazits

Die deutsche Sprache ist bekanntlich keine einfache Sprache. Und so war es nicht trivial, unsere bestehenden Blog-Beiträge so anzupassen, dass sie "gegendert" werden können. In den bisherigen Beispielen haben wir nur Begriffe betrachtet, die im Nominativ vorliegen, also "der Autor, die Autorin, die Autoren, die Autorinnen usw.". Was aber, wenn die Begriffe im Genitiv oder Dativ vorkommen, also "des Autors, der Autorin, an den Autor, an die Autorin usw."? Was ist z.B. mit so einem Satz "Einen Brief an den Autor schreiben" der, wie im Netz gesehen, so gegendert wird "Einen Brief an den*die Autor*in schreiben"? In einem femininen Kontext würden mit unserer technischen Lösung daraus "Einen Brief an **den** Autorin schreiben", was mich im Lesefluss vielleicht weniger als im Beispiel aus dem Netz, aber dennoch stören würde, einfach weil es grammatikalisch falsch ist.

Für kreative Texte, wie in Blogs oder Zeitungsartikeln, brauchen wir hier vielleicht noch erweiterte Lösungsansätze technologischer Art. **Oder aber wir gewöhnen uns einen anderen Schreibstil an.** Meine Kollegin hat das für unsere bestehenden Blogbeiträge versucht und wird demnächst hier bei uns im Blog über ihre Erfahrungen berichten.

Im Umfeld der technischen Dokumentation allerdings geht der Trend, auch wegen Übersetzungskosten, mehr und mehr zum vereinfachten Schreiben wie z.B. "Brief an Autor schreiben", also ganz ohne Beachtung der Gender-Diskussion. Es lassen sich dennoch bestehende Konzepte bestens kombinieren und mit modernen und technischen Mitteln kleinere und größere Probleme lösen. Vieles wird man allerdings nicht angehen, weil der Aufwand im Verhältnis zum Nutzen nicht gerechtfertigt scheint.

Mit unserem Lösungsansatz haben wir gezeigt, dass **aktuelle Probleme mit Technologie und vor allem mit Standards gelöst werden können.** Wir haben keine neuen Standards entwickelt, keine neue Software dafür eingeführt, sondern einfach das genutzt, was heute schon und seit sehr vielen Jahren verfügbar und im Einsatz ist. Auch wenn wir uns hier auf TBX und DITA konzentriert haben, lässt sich dieser Ansatz ebenfalls in andere Systeme portieren, egal ob Sie mit einem CCMS, mit DocBook oder gar unstrukturiert mit Word oder FrameMaker arbeiten.

Haben Sie das Thema für sich bereits gelöst? Oder haben Sie Ideen, die den oben beschriebenen Lösungsansatz unterstützen und weiterbringen? Wir freuen uns auf einen Austausch mit Ihnen über die konzeptionellen und technischen Aspekte der Genderthematik.

Über Markus Wiedenmaier

Markus Wiedenmaier ist Gründer der c-rex.net GmbH, Hersteller eines cloudbasierten Digital Transformation and Information Delivery Service. Seit 2001 arbeitet Markus als Trainer, Entwickler und Systemintegrator im Bereich XML-Publishing. Sein Fokus liegt auf halb- und vollautomatischen XML-Publishing-Lösungen, Prozessintegration, Prozessautomatisierung, Datenmodellierung und Datenmigration.