XML-Workshop für Editionsprojekte

Christian Sonder M. A.

Wiss. Mitarbeiter SSRQ, Universität St. Gallen

Trilog-Verlag für Kunst, Literatur und Wissenschaft

Mail: christian.sonder@unisg.ch oder christian.sonder@trilog-verlag.de

Was wir in der letzten Sitzung gelernt haben

2. Sitzung

1) XML-Syntax, Teil 2: Escapen, Kommentare, CDATA Sections, XML-Deklarationen.
2) Wie ist ein vollständiges XML-Dokument aufgebaut?
3) Wie kodiert man einfache Daten mit XML?
4) XML und Unicode, Character References.

Was wir in dieser Sitzung lernen

3. Sitzung

1) Was ist die Text Encoding Initiative (TEI)?
2) Was ist korrektes XML? Wohlgeformtheit vs. Validität.
3) XML-Syntax, Teil 3: Processing Instructions.
4) Verknüpfung von XML-Dateien mit Stylesheets und Schemata.
5) Umgang mit dem Oxygen XML-Editor.

Die Text Encoding Initiave (TEI)

- XML ist vielfältig einsetzbar und beliebig erweiterbar. Jeder Sachverhalt kann auf zahlreiche Arten kodiert werden. Daher sind «frei kodierte» Dateien kaum miteinander vergleichbar.
- Ohne einen Standard oder eine Handlungsempfehlung müsste jedes Editionsprojekt, das sich mit der Kodierung von Texten beschäftigt, von Grund auf eine eigene Liste von Elementen, Attributen, Attributwerten etc. überlegen, d. h. das Rad neu erfinden.
- Daher kam es 1987 (noch vor Erfindung von XML) zur Gründung der TEI, welche seit 1990 Richtlinien für die Kodierung von Texten mit Auszeichnungssprachen wie SGML (Vorgänger von XML) und XML herausgibt: https://tei-c.org/
- Die TEI-Richtlinien sind kein technischer Standard, sondern eher eine Sammlung von Empfehlungen und Vorschlägen.

Die Text Encoding Initiave (TEI)

- XML ist vielfältig einsetzbar und beliebig erweiterbar. Jeder Sachverhalt kann auf zahlreiche Arten kodiert werden. Daher sind «frei kodierte» Dateien kaum miteinander vergleichbar.
- Ohne einen Standard oder eine Handlungsempfehlung müsste jedes Editionsprojekt, das sich mit der Kodierung von Texten beschäftigt, von Grund auf eine eigene Liste von Elementen, Attributen, Attributwerten etc. überlegen, d. h. das Rad neu erfinden.
- Daher kam es 1987 (noch vor Erfindung von XML) zur Gründung der TEI, welche seit 1990 Richtlinien für die Kodierung von Texten mit Auszeichnungssprachen wie SGML (Vorgänger von XML) und XML herausgibt: https://tei-c.org/
- Die TEI-Richtlinien sind kein technischer Standard, sondern eher eine Sammlung von Empfehlungen und Vorschlägen.

Die TEI-Richtlinien

- Aktuellste Richtlinien P5 Version 4.8.0 vom 2.9.2024: 587 Elemente, 274 Attribute, ca. 2500 Druckseiten Dokumentation.
- Online unter: https://tei-c.org/release/doc/tei-p5-doc/de/html/index.html
- Die TEI-Richtlinien sind auch in Form vom XML-Schemata umgesetzt, sodass man eine XML-Datei auf Übereinstimmung mit einem TEI-XML-Schema überprüfen kann.
- Die TEI-Richtlinien sind nicht statisch, sondern dynamisch. Sie werden laufend ergänzt und erweitert und lassen sich auch durch Nutzer modifizieren.
- Vorteile: Man muss nicht bei Null starten und das Rad neu erfinden. Alle Fachrichtungen und Editionsansätze können umgesetzt werden.
- Nachteil: Die Vergleichbarkeit der Daten ist nur eingeschränkt möglich. TEI muss auf die eigenen Bedürfnisse angepasst werden.

Wohlgeformtheit vs. Validität

- Wir haben bisher nur von korrektem bzw. nicht korrektem XML gesprochen. Das kann jetzt differenziert werden:

  Definition: wohlgeformt

  Ein Datenobjekt ist wohlgeformt, wenn es den Regeln der XML-Syntax entspricht.

  Definition: valide

  Ein Datenobjekt ist valide, wenn es wohlgeformt ist und zusätzlich den Anforderungen einer Document Type Definition (DTD) oder eines Schemas entspricht.

- Wir können also zwei Arten von Korrektheit unterscheiden: Ist ein Dokument wohlgeformt, dann ist es in Sinne der XML-Syntax korrekt. Ist ein Dokument valide, dann ist es nicht nur wohlgeformt, sondern auch korrekt im Sinne eines Schemas.

Wohlgeformtheit vs. Validität

- <person>Hektor</Person> ist nicht wohlgeformt. Weil Anfangs- und Endtag nicht übereinstimmen. Damit ist der Ausdruck kein XML.
- <person>Hektor</person> ist wohlgeformt. Damit ist der Ausdruck korrektes XML. Ob dieser Ausdruck auch valide ist, hängt davon ab, ob es eine DTD oder ein Schema gibt, welches das Element person mit Text als möglichem Inhalt definiert.
- Daher ist es immer sinnvoll, eine XML-Datei direkt mit einem Schema zu verknüpfen. Die Verknüpfung erfolgt mithilfe einer Processing Instruction.

Processing Instructions

- Processing Instructions geben einem XML-Prozessor Anweisungen, wie mit einem XML-Dokument umgegangen werden soll.
- Die beiden häufigsten Anwendungen sind die Verknüpfung eines XML-Dokuments mit einem Schema oder mit einem Stylesheet.
- Eine Processing Instruction besteht aus:

  <? + Name der Instruction + die Instruction selbst + ?>

- Wichtige Beispiele:

```
<?xml-stylesheet type="text/css" href="style.css"?>      [Link zu einer CSS-Datei]
<?xml-stylesheet type="text/xsl" href="style.xsl"?>      [Link zu einer XSL-Datei]
<?xml-model href="http://www.tei-c.org/.../schema/relaxng/tei_all.rng"
                    type="application/xml"?>             [Link zu einem RNG-Schema]
```

Übung 1: Processing Instructions

- Öffnen den Kurs auf GitHub und laden Sie die Dateien hektor.xml, hektor.css und hektor.xsd herunter.
- Öffnen Sie die Datei hektor.xml mit Ihrem Browser.
- Fügen Sie die folgende Processing Instruction hinzu und aktualisieren Sie Ihren Browser:
  <?xml-stylesheet type="text/css" href="hektor.css"?>
- Öffnen Sie die Datei hektor.xml mit dem Oxygen und drücken Sie auf Validieren.
- Fügen Sie die folgende Processing Instruction hinzu und drücken Sie erneut auf Validieren:
  <?xml-model type="application/xml" href="hektor.xsd">

Der komplette Kurs auf GitHub

https://github.com/ChristianSonderUniSG/xml-kurs

Dort finden Sie alle Folien, Übungsaufgaben, Hausaufgaben und weitere Materialien.