

# **Kollationen und Transformationen für XML–Dokumente**

Dietmar Seipel, Klaus Prätor

Universität Würzburg,  
Berlin–Brandenburgische Akademie der Wissenschaften

# Kollation von XML-Dokumenten

Michael Wenger hat einen XML-Vergleichsalgorithmus für die philologische Arbeit als Eclipse-Plugin entwickelt.

- Elemente werden basierend auf einem *Ähnlichkeitsmaß* zugeordnet.
- Exakte Übereinstimmungen werden mittels einer Hashsumme ermittelt; falls der Textinhalt gleich geblieben ist, aber neue Auszeichnungen hinzugekommen sind, so kann dies gut mit Hilfe einer *toleranten Hashsumme* festgestellt werden.

# Kollation von XML-Dokumenten

- Verfeinerung des Grundalgorithmus durch beliebige *Ähnlichkeitsfilter* und *Strukturfilter*.
- Falls ein Strukturfilter zwei Knoten einander zuordnet, so wird auch versucht die Nachbarknoten innerhalb eines eingeschränkten Teilbereichs des zweiten Dokuments zuzuordnen.
- Die Reihenfolge der Knoten innerhalb des Dokuments wird dabei berücksichtigt.

# Kollation von XML-Dokumenten

- Elemente sind aber nicht so stark an ihre Eltern gebunden wie bei klassischen XML-Daten, so daß *Elementverschiebungen* erkannt werden können. Außerdem werden Elementnamen und Attribute nicht so stark gewichtet, so daß in einem gewissen Umfang sogar *geänderte Strukturauszeichnungen* erkannt werden können.
- Als Vorlage für die Ausgabe der *Differenzlisten (Deltas)* zwischen den beiden Dokumenten wird das Ausgabeformat von *XyDiff* benutzt.

# Kollation von XML-Dokumenten – Beispiel

```
<p n="40">  
  <s n="1">  
    Der Igel ist vor dem Hasen.  
  </s>  
  <s n="2">  
    Der Hase isst schneller  
    als der Igel.  
  </s>  
  <s n="3">  
    Der Igel gewinnt dennoch.  
  </s>  
  <s n="4">  
    Sehr seltsam.  
  </s>  
</p>
```

```
<p n="50">  
  <s n="1">  
    Der Hase ist schneller  
    als der Igel.  
  </s>  
  <s n="2">  
    Trotzdem ist der Igel  
    vor dem Hasen.  
  </s>  
</p>  
<p n="51">  
  <s n="1">  
    Der Igel gewinnt dennoch.  
  </s>  
</p>
```

# Transformation von XML-Dokumenten

- Kritische oder wissenschaftliche Editionen sind ein vielversprechendes Anwendungsfeld für die *deklarative Programmierung*. Diese kann das Parsen und das Markup von Texten sowie die Transformation von XML-Dokumenten vereinfachen.
- FNQUERY ist eine deklarative XML-Anfrage- und Transformationssprache, welche voll verschränkt ist mit der Programmiersprache PROLOG.
- Die GUI von FNQUERY ermöglicht die *inkrementelle* Entwicklung von Transformationsmethoden.

# Nützliche Features von PROLOG

- Spracherweiterungen: Define a *little language* embedded in PROLOG syntax (O'Keefe).
- Definite Clause Grammars (DCGs)
- Backtracking und Unifikation
- Prädikate höherer Ordnung:
  - Möglichkeit alle Resultate zu berechnen
  - Iteratoren und Filter

# Orthogonale Transformationstechniken

Wir präsentieren einen neuen, kompakten Substitutionsformalismus, der mit den bekannten DCGs von PROLOG verwoben werden kann:

- *DCGs zur Gruppierung* von Elementen auf derselben Ebene des Dokuments und komplexe, verschachtelte Strukturen zu bilden (sequentieller Scan).
- *Substitutionsregeln* zur Transformation komplexer, verschachtelter Dokumente; sie transformieren ein (baumstrukturiertes) XML-Dokument rekursiv – mit den Blättern beginnend.



# Parsing und Transformation

```
<p>Samstag, 27.05.</p>
```

```
<p><em>Das Projekt TextGrid</em></p>
```

```
<p><em>Kollationierung und Transformation</em></p>
```

## Parsing:

```
<section title="Samstag, 27.05.">
```

```
  <p><em>Das Projekt TextGrid</em></p>
```

```
  <p><em>...</em></p>
```

```
</section>
```

## Transformation:


```
<Tag Datum="Samstag, 27.05.">
```

```
  <Vortrag>Das Projekt TextGrid</Vortrag>
```

```
  <Vortrag>...</Vortrag>
```

```
</Tag>
```

# Kommentar zu einem Brief aus der Jean Paul–Edition



The screenshot shows a Mozilla browser window with the following elements:

- Menu Bar:** File, Edit, View, Go, Bookmarks, Tools, Window, Help
- Navigation Bar:** Back, Forward, Home, Stop, Address Bar (Jean\_Paul/JPK001.xml), Search, Print, and a logo.
- Bookmark Bar:** Home, Bookmarks, The Mozilla Or..., SuSE - Linux
- Document Content:**
  - Section Header:** 1. Von Erhard Friedrich Vogel. Rehau, 6. Mai 1781, Sonntag
  - Section Header:** Überlieferung
  - Text:** H: BL, Eg, 2008. 1 Bl. 2°, 1/2 S.  
Erster Druck: II 1, XXIII (Einleitung zu den Übungen im Denken) (unvollständig).  
Beilagen: Vogels Anmerkungen zu Jean Pauls Aufsatz ...
  - Section Header:** Erläuterungen
  - Text:** Erhard Friedrich Vogel, am 17. November 1750 als ältester Sohn des Bayreuther Hofkammerrates Johann Achatius Vogel und seiner Frau Anna Elisabetha geb. Niedermann geboren, gestorben am 2. Mai 1823 als Dekan in Wunsiedel, war Mentor und Freund Jean Pauls in dessen Jugendjahren und in der Zeit des mühsamen Aufbaus einer schriftstellerischen Existenz. ...
  - List-Group:**
    - 1 Am Ende seiner Schulzeit ...
    - ...
    - 7 Dem folgenden Brief Vogels ist zu entnehmen, ...
- Status Bar:** Document: Done (0.1 secs)

## Konversion: Word $\mapsto$ HTML

```
<p style="margin-top:0;margin-bottom:0;">
  <font face="Times New Roman" size="3">
    <em>202, </em> 6
    <strong>Krebse] </strong>
    Buchhaendlerisch: Remittenden.
  </font>
</p>
```

# Konversion: HTML $\mapsto$ XML / TEI

- die XML-Anfragesprache FNQUERY

- Pfadausdrücke  $\rightsquigarrow$  FNPATH
- Selektion von Elementen/Attributen  $\rightsquigarrow$  FNSELECT
- Transformationen wie in XSLT  $\rightsquigarrow$  FNTRANSFORM
- Updates  $\rightsquigarrow$  FNUPDATE

- Parsing mittels DCGs

- graphische Benutzeroberfläche (GUI)

# Semantisches Tagging: HTML $\mapsto$ XML

```
<comment>
  <commentHead>1. Von Erhard Friedrich Vogel.
    Rehau, 6. Mai 1781, Sonntag
  </commentHead>
  <ednote type="Ueberlieferung">
    <notep>H: BL, Eg. 2008. 1 Bl. 2, 1/2 S.</notep> ...
  </ednote>
  <ednote type="Erlaeuterungen"> ...
    <notep>
      <page>202, </page> 6
      <lemma>Krebse] </lemma>
      Buchhaendlerisch: ...
    </notep>
  </ednote>
</comment>
```

# Die GUI von FNQUERY (FNTRANSFORM)

**Source Directory:**

**Source File:**  ▼

```
<?xml version='1.0'
  encoding='ISO-8859-1' ?>

<a>
  <b u="1" v="2"/>
  <c w="3"/>
  <d x="4"/>
</a>
```

```
a:Es1 ---> a:[]:[E|Es2] :-
  bc(E, Es1, Es2).

d:AsEs ---> e:AsEs.

T:AsEs ---> T:AsEs.

bc(bc:As:[]) -->
  [b:As1:[], c:As2:[],
  { append(As1, As2, As) }.
```

```
<a>
  <bc u="1" v="2" w="3"/>
  <e x="4"/>
</a>
```

# Triple-Representation von XML-Elementen

```
p:[style:'...']:[  
  font:[face:'...', size:'3']:[  
    em:['202, ', '6', ...]]
```

T:As:Es

```
<p style="...">  
  <font face="..." size="3">  
    <em>202, </em> 6 ...  
  </font>  
</p>
```

# Pfadausdrücke in FNUPDATE

```
xml_extract_notes(Xml_1, Xml_2) :-  
    Xml_2 := Xml_1 <-> [  
        ^ednote::[@type='Erlaeuterungen']  
        ^note^remark ].
```

```
<ednote type="Erlaeuterungen">  
  <note id="7">  
    <remark>  
      <pos page="1" line="18-19"/>  
      <lemma>Gehen</lemma>  
      <lemma>sind</lemma>  
    </remark>  
    Dem folgenden Brief ...  
  </note>  
</ednote>
```

```
<ednote type="Erlaeuterungen">  
  <note id="7">  
    Dem folgenden Brief ...  
  </note>  
</ednote>
```



# Zusammenfassung

- Interessant wäre ein Vergleich der verschiedenen Kollationierungsansätze (Wenger, Hemmerich, etc.)
- FNQUERY ist eine deklarative Anfrage-, Transformations- und Programmiersprache.
- Wir können das Parsen und Transformieren von XML-Dokumenten mischen.
- Vor- und Nachbearbeitung bei der Kollationierung