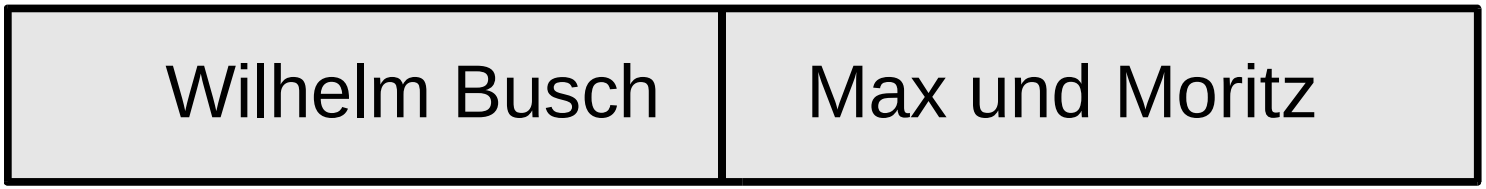
Separatoren und textuelle Darstellung

<draheim@informatik.hu-berlin.de>

Dezember 2002

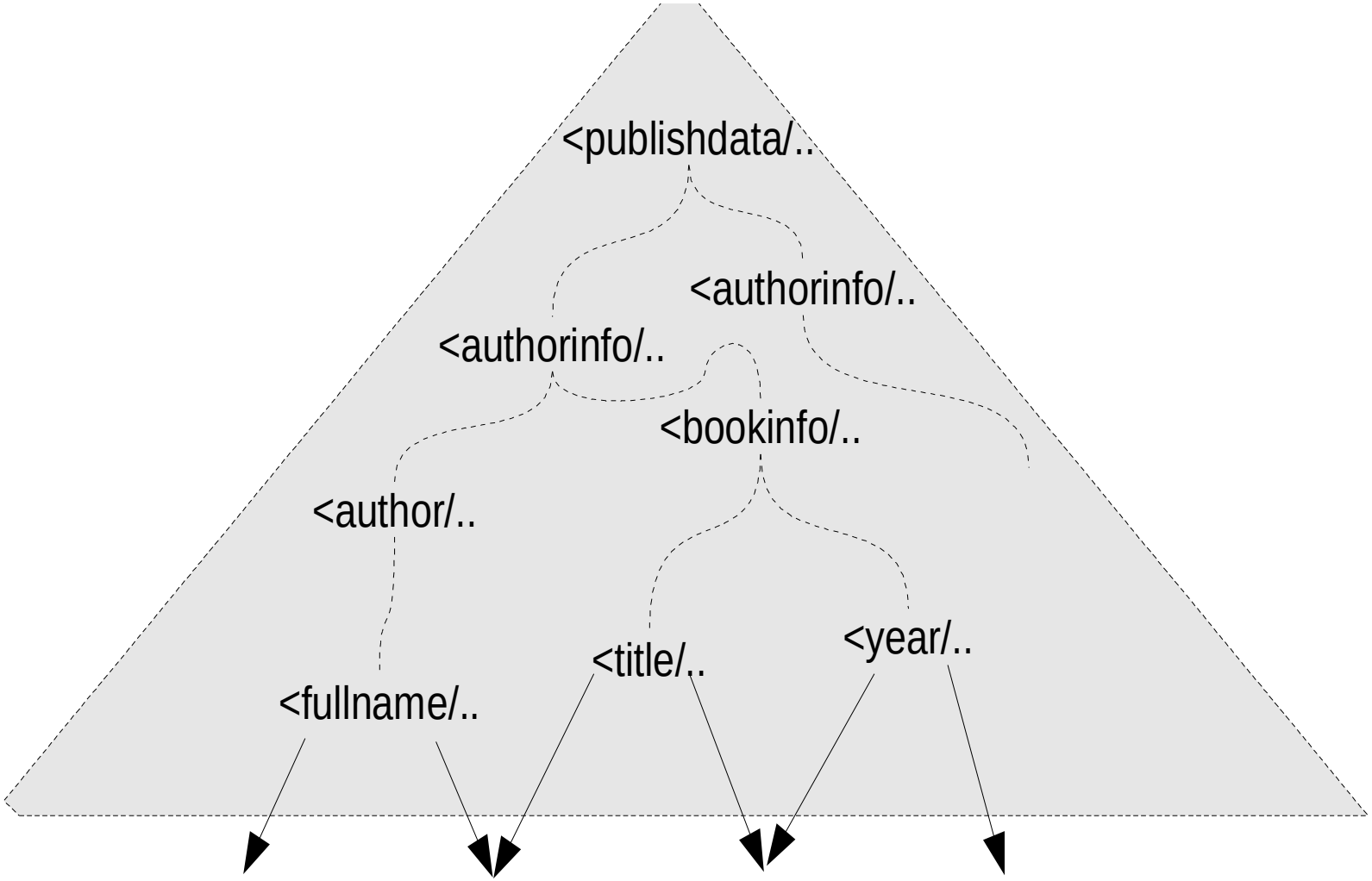


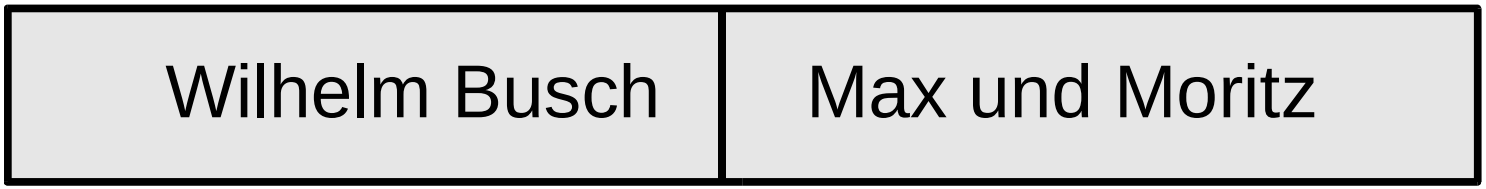
<author> <title>

<author>Wilhelm Busch</author><title>Max und Moritz</title>

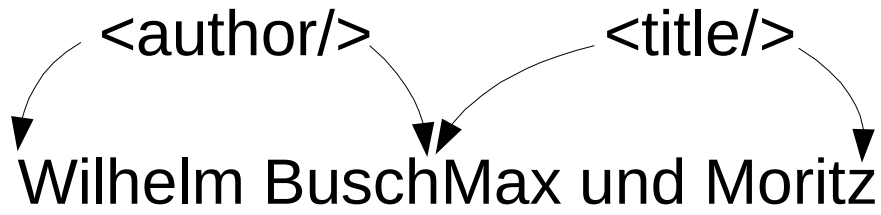
```
<?xml version=“1.0“ encoding=“UTF-8“ ?>
<!DOCTYPE book SYSTEM “book.dtd“>
<book>
  <record>
    <author>Wilhelm Busch</author>
    <title>Max und Moritz</title>
  </record>
</book>
```

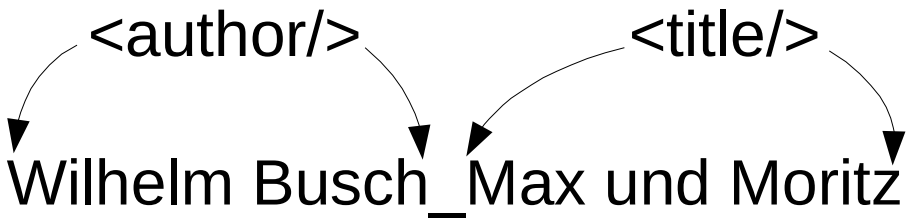
AST/TA – Baum


<publishdata/..
<authorinfo/..
<authorinfo/..
<bookinfo/..
<author/..
<year/..
<title/..
<fullname/..
Wilhelm BuschMax und Moritzim Jahre 1865



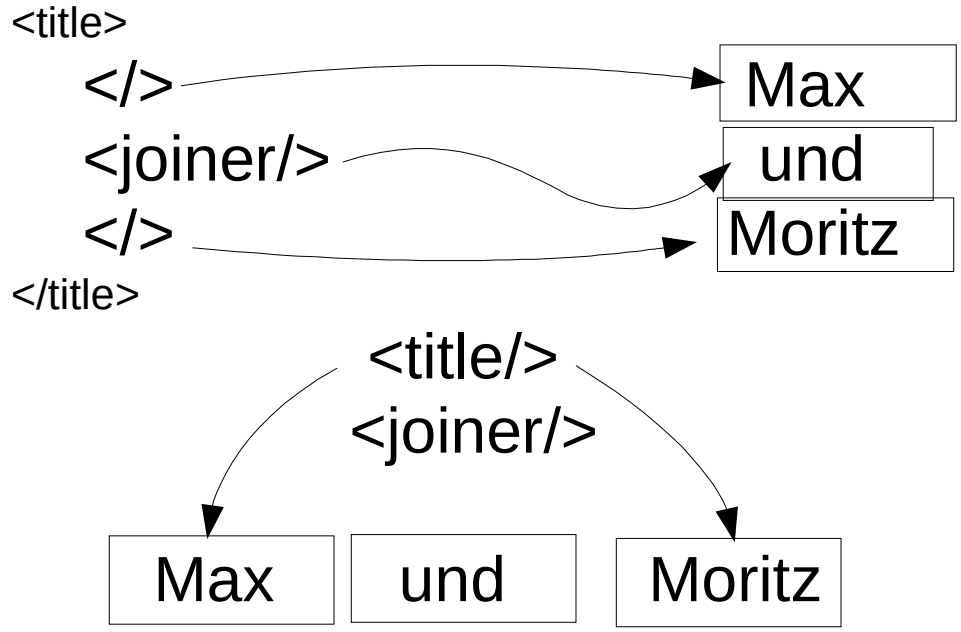
<author>Wilhelm Busch</author><title>Max und Moritz</title>





vollständiger DOM:

<title>Max <joiner>und</joiner> Moritz</title>



suche nach "Max und Moritz"

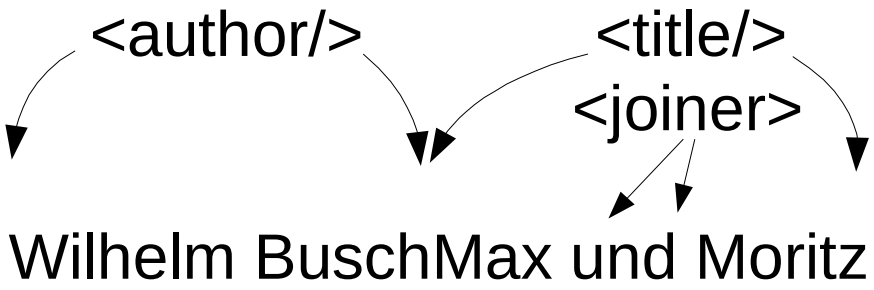
textabschnitte in feldern:

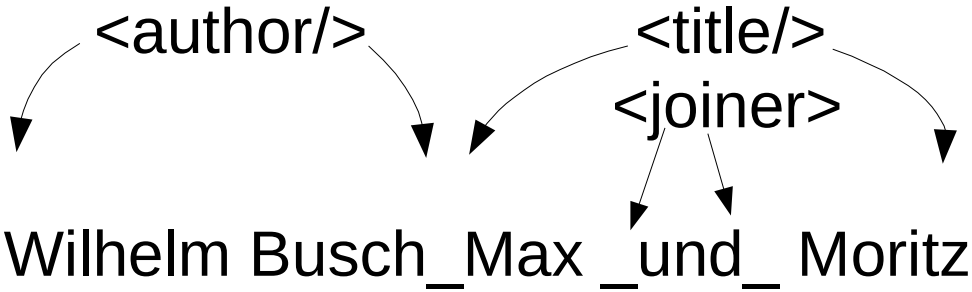
<word><big>X</big>ML</word>

und wo ist XML ?

Wilhelm Busch	Max <joiner>und</joiner> Moritz

<author>Wilhelm Busch</author><title>
Max <joiner>und</joiner> Moritz</title>

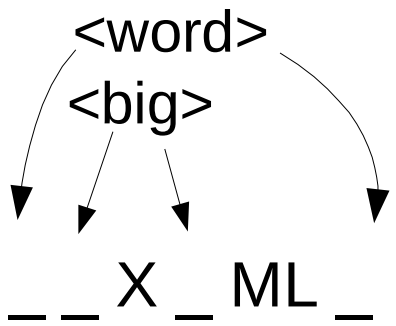




textabschnitte in feldern:

<word><big>X</big>ML</word>

und wo ist XML ?



PCRE: */*XML*/* **...** (angepasster Algorithmus, der Separatoren überspringt?)

hinweise:

- unicode kennt zero-width space (bzw. joiner)
- libpcre kennt utf-8 kodierung
- PCRE spec kennt **'\b'** als zero-width match

anpassung leicht?! *(fetchnext-backtrack)*

GUT:

- anpassung leicht, zumindest leichter als <...> live parsing

ABER:

- einfügen neuer markups!

  ohne separatoren muss nur der AST angepasst werden, mit ihnen, auch der TA
  vorsicht längenänderung! positionen im AST nachführen? oder den TA vollständig punktieren, dass zwischen zwei zeichen elemente immer ein separator passt, oder separatoren einfach nicht mitzählen, und der TA diesen beachten fakt kann.

- anpassen aller algorithmen!

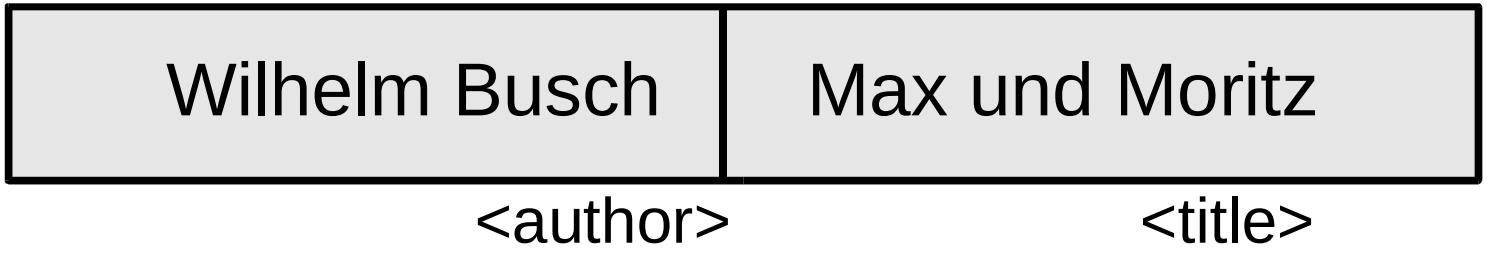
  bei text mag das ja leicht sein, aber es gibt auch andere datenstrukturen, bei denen zugriff in die unterliegenden datenstrukturen sich etwas schwieriger gestaltet, die sich aber leicht für das AST/TA konzept eignen würden – ist AST/TA auf "text"-mining beschränkt?

- wann sollen separatoren implizit übergangen werden, und wann sollen sie eine suche abgrenzen? (fetchnext)

  man nehme verschiedene typen von separatoren? wieviele?
  erweiterung der suchfragen, welche separatoren abgrenzen sollen!
  vergleiche dazu PCRE /m und /s modifier!

ähm, ist das die richtige frage?

DB generierung nimmt an, dass zeichen im text "sichtbar" sind!



<author>Wilhelm Busch</author><title>Max und Moritz</title>

```
<?xml version=“1.0“ encoding=“UTF-8“ ?>
<!DOCTYPE book SYSTEM “book.dtd“>
<book>
  <record>
    <author>Wilhelm Busch</author>
    <title>Max und Moritz</title>
  </record>
</book>
```

w3.org/TR/REC-CSS2/visuren.html ..... display property

<A>a<B><C>b</C></B>c</A>

C { display : inline }

abc

C { display : block }

a
b
c

B { display : compact }

a
b
c

immer!

nie:

a

b

c

9.3.2 if a block box follows a compact box then it behaves like a one-line inline box, otherwise the compact box becomes a block box

und mit separatoren?

abc

a.b.c immer existent, aber

a..b..c mal impl übergangen?

- markup attribute bestimmt visualizierung
- was ist "visueller" text – vgl. DB
- markup attribut bestimmt (aktuelle) abgrenzung
- nicht teil der xml syntax selbst!
- separierung auch abhängig von kontext
- abhängig von daten schachtelung...

(attribut mitwirkung bei einlesen, umformatierung bei änderung?)

vorteil: beim einlesen wird schlicht geprüft, dass die separatoren für operationen passend sind.

Vorteil:

- das AST/TA ermöglicht, schon bekannte Operationen zu nehmen, die auf den Datenbereichen sonst selbst arbeiten, so braucht man keine Anpassung an DOM oder DB/xml Implementation selbst.
- spezialisierte Implementationen mit native XML verwendung
- Umsetzen von einem XML baum in den nächsten ist simpel.