News from boxes and glue

TUG 2023
July 14 2023
Hotel Collegium Leoninum
Bonn

Patrick Gundlach
gundlach@speedata.de

speedata
Berlin, Germany
Lessons learned

TUG 2023
July 14 2023
Hotel Collegium Leoninum
Bonn

Patrick Gundlach
gundlach@speedata.de

speedata
Berlin, Germany
What is “boxes and glue” (recap)?

boxes and glue ...
What is “boxes and glue” (recap)?

boxes and glue ...

... is a collection of software libraries
What is “boxes and glue” (recap)?

boxes and glue ...

... is a collection of software libraries

... not a ready-to-run piece of software
What is “boxes and glue” (recap)?

boxes and glue ...

... is a collection of software libraries

... not a ready-to-run piece of software

... written in the Go programming language
What is “boxes and glue” (recap)?

boxes and glue ...

... is a collection of software libraries

... not a ready-to-run piece of software

... written in the Go programming language

... the attempt to bring TEX's superb typesetting quality to a modern environment
What is “boxes and glue” (recap)?

boxes and glue ...

... is a collection of software libraries
... not a ready-to-run piece of software
... written in the Go programming language
... the attempt to bring TEx's superb typesetting quality to a modern environment
... and of course OpenSource
Why boxes and glue?

(Fully automatic) catalog production with LuaTEX
Same data structures and API
Same data structures and API

- Nodes (glyph, glue, rule, whatsit)
Same data structures and API

- Nodes (glyph, glue, rule, whatsit)
- Node packing (vpack, hpack)
Same data structures and API

- Nodes (glyph, glue, rule, whatsit)
- Node packing (vpack, hpack)
- Rectangular items
Same data structures and API

- Nodes (glyph, glue, rule, whatsit)
- Node packing (vpack, hpack)
- Rectangular items
- Line breaking
Same data structures and API

- Nodes (glyph, glue, rule, whatsit)
- Node packing (vpack, hpack)
- Rectangular items
- Line breaking
- Hyphenation
Same data structures and API

- Nodes (glyph, glue, rule, whatsit)
- Node packing (vpack, hpack)
- Rectangular items
- Line breaking
- Hyphenation
- ...

Boxes & glue
Architecture of boxes and glue
Architecture of boxes and glue
Architecture of boxes and glue
Architecture of boxes and glue

Frontend
- Font families
- Colors
- CSS/HTML
- Interaction
- Accessibility
- Page layout

Backend
- Fonts
- Language
- Document/Pages
- Images
- Nodes
- PDF library
Architecture of boxes and glue
Architecture of boxes and glue

Frontend:
- Font families
- Colors
- CSS/HTML
- Interaction
- Accessibility
- Page layout

Backend:
- Fonts
- Language
- Document/Pages
- Images
- Nodes
- PDF library
Architecture of boxes and glue

Frontend:
- Font families
- Colors
- CSS/HTML
- Interaction
- Accessibility
- Page layout

Backend:
- Fonts
- Language
- Document/Pages
- Images
- Nodes
- PDF library
Architecture of boxes and glue

HTML/CSS typesetting
- Application
- Catalog software (XML based)

Frontend
- Font families
- Colors
- CSS/HTML
- Interaction
- Accessibility
- Page layout

Backend
- Fonts
- Language
- Document/Pages
- Images
- Nodes
- PDF library
Time line

**raphink** on Sep 8, 2016
Are you going to port LuaTex to Go? 😊

**pgundlach** on Sep 8, 2016  Author  Member
It's a long term project and only parts of TeX will be ported (only the algorithms and node stuff, but not the input language)

**raphink** on Sep 8, 2016
The input language is what I'd love to see disappear in TeX. After 10 years of using it, I still get headaches from trying to write simple logic in TeX packages. I'd love a new language with real modern expressivity.
Time line
Time line

web → go experiments
Time line

web → go experiments

first ТEx algorithm
**Time line**

- web → go experiments
- first \( \text{T}_\text{E}X \) algorithm
- CSS/HTML typesetting experiments
Time line

web → go experiments
first \textsc{TeX} algorithm
CSS/HTML typesetting experiments
first commit boxes and glue

Time line

- web → go experiments
- first \( \text{TeX} \) algorithm
- CSS/HTML typesetting experiments
- Lua frontend
- first commit boxes and glue
Time line

- web → go experiments
- first TeX algorithm
- CSS/HTML typesetting experiments
- first commit boxes and glue
- Lua frontend
- XML XPath 2 library

Time line

web → go
experiments
first TEX
algorithm
CSS/HTML
typesetting
experiments
first commit
boxes and glue
Lua
frontend
XML
XPath 2
library
XTS

Time line

- web → go experiments
- first TEX algorithm
- CSS/HTML typesetting experiments
- Lua frontend
- first commit boxes and glue
- CSS/HTML typesetting
- XTS
- XML XPath 2 library

Next steps / wishes
Next steps / wishes

- Right to left (and mixed) typesetting
Next steps / wishes

- Right to left (and mixed) typesetting
- Graphics library like MetaPOST
Next steps / wishes

- Right to left (and mixed) typesetting
- Graphics library like MetaPOST
- Paragraph shape
Next steps / wishes

- Right to left (and mixed) typesetting
- Graphics library like MetaPOST
- Paragraph shape
boxes and glue: design goals
boxes and glue: design goals

- TeX alike typography and output quality
boxes and glue: design goals

- \(\TeX\) alike typography and output quality
- Performance
boxes and glue: design goals

- \( \text{T\,E\,X} \) alike typography and output quality
- Performance
- \( \text{T\,E\,X}'s \) data structures
boxes and glue: design goals

- T\TeX alike typography and output quality
- Performance
- T\TeX's data structures
- Arabic et. al. (Unicode, LTR/RTL, Bidi)
boxes and glue: design goals

- \( \text{T\TeX} \) alike typography and output quality
- Performance
- \( \text{T\TeX}'s \) data structures
- Arabic et. al. (Unicode, LTR/RTL, Bidi)
- PDF standards
boxes and glue: design goals

- TEX alike typography and output quality
- Performance
- TEX's data structures
- Arabic et. al. (Unicode, LTR/RTL, Bidi)
- PDF standards
Lessons learned (so far)
Lessons learned (so far)

- It works!
Lessons learned (so far)

- It works!
- Writing PDF can be difficult
Lessons learned (so far)

- It works!
- Writing PDF can be difficult
- Err and err and err again
It works!

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful; and the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain; and when she was bored she took a golden ball, and threw it up on high and caught it; and this ball was her favorite plaything.

The frog king

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face.

Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

1. The king's daughter followed it with her eyes, but it vanished, and the well was deep, so deep that the bottom could not be seen.
2. At this she began to cry, and cried louder and louder, and could not be comforted.
Lessons learned (so far)

- It works!
- Writing PDF can be difficult
- Err and err and err again
Writing PDF can be difficult
Writing PDF can be difficult

- Looks like an innocent file format
  (plain text + some binary data)
Writing PDF can be difficult

- Looks like an innocent file format (plain text + some binary data)
- Spec has more than 1000 pages
“Innocent file format”

5 0 obj
<<
/Type /XObject
/Subtype /Image
/DecodeParms << /Columns 1072 /Predictor 15 >>
/BitsPerComponent 8
/ColorSpace [/Indexed /DeviceRGB 128 6 0 R]
/Filter /FlateDecode
/Height 804
/Length 22877
/Width 1072
>>
stream
...
<compressed data>
...
endstream
endobj
“Innocent file format”
“Innocent file format”
“Innocent file format”
“Innocent file format”
“Innocent file format”
“Innocent file format”
“Innocent file format”

5+7+5=
10+7= 17
“Innocent file format”

5 + 7 + 5 = 10 + 7 = 17

5 + 7 + 5 = 10 + 7 = 17
Lessons learned (so far)

- It works!
- Writing PDF can be difficult
- Err and err and err again
Lessons learned (so far)

- It works!
- Writing PDF can be difficult
- Err and err and err again
  but not less and less and less
Different kind of errors

<table>
<thead>
<tr>
<th>Cell 1</th>
<th>Cell 2</th>
<th>Cell 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell 4</td>
<td>Cell 5</td>
<td>Cell 6</td>
</tr>
</tbody>
</table>
Different kind of errors

<table>
<thead>
<tr>
<th>Cell 1</th>
<th>Cell 2</th>
<th>Cell 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell 4</td>
<td>Cell 5</td>
<td>Cell 6</td>
</tr>
</tbody>
</table>

In olden times when here lived a king beautiful; and he had he sun himself, was as onished when
Different kind of errors

(empty PDF)
Different kind of errors

(emptyp PDF)
Writing PDF (error checking)
Writing PDF (error checking)
Writing PDF (error checking)
Writing PDF (error checking)
Tools for PDF debugging
Tools for PDF debugging

- Adobe Acrobat
Tools for PDF debugging

- Adobe Acrobat

- less
  (use qpdf to decompress):
  qpdf --qdf --object-streams=disable in.pdf out.pdf
Tools for PDF debugging

- Adobe Acrobat
- less
  (use qpdf to decompress):
  
  ```
  qpdf --qdf --object-streams=disable in.pdf out.pdf
  ```
- veraPDF (OpenSource)
Tools for PDF debugging

- Adobe Acrobat
- less
  (use qpdf to decompress):
    qpdf --qdf --object-streams=disable in.pdf out.pdf
- veraPDF (OpenSource)
- PAC (PDF Accessibility Checker) from PDF/UA foundation
Tools for PDF debugging

- Adobe Acrobat
- less
  (use qpdf to decompress):
  qpdf --qdf --object-streams=disable in.pdf out.pdf
- veraPDF (OpenSource)
- PAC (PDF Accessibility Checker) from PDF/UA foundation

Development cycle

implement feature and write PDF → open in Adobe Acrobat → check for errors
Debugging...
### Debugging...

<table>
<thead>
<tr>
<th>Cell 1</th>
<th>Cell 2</th>
<th>Cell 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell 4</td>
<td>Cell 5</td>
<td>Cell 6</td>
</tr>
</tbody>
</table>
Debugging...

- Use a step by step debugger
Debugging...

- Use a step by step debugger
- Visual debugging

A short story

A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquil existence, that I neglect my talents...
Debugging...

- Use a step by step debugger
- Visual debugging
- "Printf" debugging

```plaintext
a := readInput()
printf("input is:", a)
doSomething(a)
...
```
Debugging...

- Use a step by step debugger
- Visual debugging
- “Printf” debugging
- Rubber duck debugging
Debugging...

- Use a step by step debugger
- Visual debugging
- “Printf” debugging
- Rubber duck debugging

```java
min < math.MaxInt {
    lb.appendBreakpointHere(n, dmin, dc, dc, ...
}
if dmin == math.MaxInt & lb.activeNodesA == nil {
    W, E, Y, Z := lb.computeSum(n)
    lastInactive := lb.inactiveNodesP
    width := lb.sumW
    var pre Node
    switch v := n.(type) {
    case *Penalty:
        width += v.Width
    case *Disc:
        width += s * bag.Factor
        pre = v.Pre
```
Debugging...

- Use a step by step debugger
- Visual debugging
- "Printf" debugging
- Rubber duck debugging
Printf debugging...
Printf debugging...

- Works only for simple cases
Printf debugging...

- Works only for simple cases
- Too much output for data structures used in typesetting
Printf debugging...

- Works only for simple cases
- Too much output for data structures used in typesetting
- How do I understand the nested node lists?
Printf debugging...

- Works only for simple cases
- Too much output for data structures used in typesetting
- How do I understand the nested node lists?

Structured output is important!
Node debugging

viz nodelist
Node debugging

nodetree
(by Josef Friedrich)
Structured debugging output

<vlist id="567" wd="538.58" ht="48" dp="0" origin="textblock"/>
<vlist id="566" wd="538.58" ht="48" dp="0"/>
<hlist id="564" wd="538.58" ht="48" dp="0" r="1" origin="line"/>
<glue id="563" wd="0" stretch="0" stretchorder="0" shrink="0" shrinkorder="0" subtype="0"/>
<vlist id="556" wd="538.58" ht="48" dp="0"/>
<vlist id="555" wd="538.58" ht="48" dp="0" origin="prepend in HTML mode" x="0" y="0"/>
<hlist id="553" wd="538.58" ht="8.02" dp="1.98" r="15.671976806286716" origin="line"/>
<glue id="552" wd="0" stretch="0" stretchorder="0" shrink="0" shrinkorder="0" subtype="0"/>
<glyph id="4" components="L" wd="5.56" ht="8.02" dp="1.98" codepoint="72" face="0"/>
<glyph id="5" components="o" wd="5.56" ht="8.02" dp="1.98" codepoint="82" face="0"/>
<glyph id="6" components="r" wd="3.33" ht="8.02" dp="1.98" codepoint="97" face="0"/>
<kern id="7" kern="-0.1"/>
...
<glyph id="482" components="." wd="2.78" ht="8.02" dp="1.98" codepoint="89" face="0"/>
<penalty id="533" penalty="10000" width="0"/>
<glue id="534" wd="0" stretch="1" stretchorder="1" shrink="0" shrinkorder="0" subtype="0"/>
<glue id="539" wd="165.58" stretch="1" stretchorder="3" shrink="0" shrinkorder="0" subtype="0"/>
<hlist>  
<glue id="554" wd="2" stretch="0" stretchorder="0" shrink="0" shrinkorder="0" subtype="0" origin="last lineskip"/>
</hlist>  
</vlist>  
<penny id="557" penalty="10000" width="0"/>
<glue id="558" wd="1" stretch="1" stretchorder="1" shrink="0" shrinkorder="0" subtype="0"/>
<glue id="562" wd="0" stretch="0" shrink="0" shrinkorder="0" subtype="0"/>
<hlist>  
<glue id="565" wd="0" stretch="0" stretchorder="0" shrink="0" shrinkorder="0" subtype="0" origin="last lineskip"/>
</hlist>  
</vlist>
Next steps (from last year)
Next steps (from last year)

- Experiment with the algorithms
Next steps (from last year)

- Experiment with the algorithms
- Optimizations for page break and paragraph break
Next steps (from last year)

- Experiment with the algorithms
- Optimizations for page break and paragraph break
- Parallel tasks
Conclusion (1)
Conclusion (1)

- Porting \TeX\ algorithms and data structures is possible
Conclusion (1)

- Porting \TeX\ algorithms and data structures is possible
- Development takes much more time than estimated
Conclusion (1)

- Porting T\(\text{E}\)X algorithms and data structures is possible
- Development takes much more time than estimated
- Seeing the results keeps my motivation high
Conclusion (2)

“Roses are red
Violets are blue
I create my PDF
With boxes and glue

Homepage https://boxesandglue.dev

GitHub https://github.com/speedata/boxesandglue

Mastodon @boxesandglue@typo.social