The stony road to complex page layout
(and its road blocks)

フランク・ミッテルバッハ

2013-10-23
Translated . . .

When typesetting complex layouts is not fully up to the task (unless we restrict the scenario)
Instead
The stony road to complex page layout

Frank Mittelbach

The Ask

Some History

TEX’s Model

User Wishlist

Balancing & Floats
Columns
Column dependency
Rules & Backgrounds
At the border
Wishlist summary

Typesetting Challenges

Balancing
Right to left
Spaces & penalties

Summary

is

needed?
Moving on ...

1. The Ask
2. Some History
3. \TeX\’s Model
4. User Wishlist
   - Balancing & Floats
   - Columns
   - Column dependency
   - Rules & Backgrounds
   - At the border
   - Wishlist summary
5. Typesetting Challenges
   - Balancing
   - Right to left
   - Spaces & penalties
6. Summary
The Ask

Some History
TEX’s Model
User Wishlist
Balancing & Floats
Columns
Column dependency
Rules & Backgrounds
At the border
Wishlist summary
Typesetting
Challenges
Balancing
Right to left
Spaces & penalties
Summary

The stony road to complex page layout

Frank Mittelbach

WANTED
Dead or Alive

REWARD OF
$250,000

BY ORDER OF THE SHERIFF
The Ask

For:

- multiple columns
- text balancing
- float positioning
- grid typesetting
- ... and much more
Conflicting goals

- Varying the number of columns on a page

- Balancing

- Automatic float placement

- Grid typesetting

- Do everything automatically
Conflicting goals

- Varying the number of columns on a page
  - what happens with footnotes?
  - what happens with floats?

- Balancing

- Automatic float placement

- Grid typesetting

- Do everything automatically
Conflicting goals

- Varying the number of columns on a page
  - what happens with footnotes?
  - what happens with floats?

- Balancing
  - but have explicit columns breaks obeyed
  - but change typesetting based on columns

- Automatic float placement

- Grid typesetting

- Do everything automatically
Conflicting goals

- Varying the number of columns on a page
  - what happens with footnotes?
  - what happens with floats?

- Balancing
  - but have explicit columns breaks obeyed
  - but change typesetting based on columns

- Automatic float placement
  - into and across columns
  - maintaining relationships to call-outs
  - and produce pleasing (and compact) layout

- Grid typesetting

- Do everything automatically
Conflicting goals

- Varying the number of columns on a page
  - what happens with footnotes?
  - what happens with floats?

- Balancing
  - but have explicit columns breaks obeyed
  - but change typesetting based on columns

- Automatic float placement
  - into and across columns
  - maintaining relationships to call-outs
  - and produce pleasing (and compact) layout

- Grid typesetting
  - but use elements of varying and unknown heights

- Do everything automatically
Conflicting goals

- Varying the number of columns on a page
  - what happens with footnotes?
  - what happens with floats?

- Balancing
  - but have explicit columns breaks obeyed
  - but change typesetting based on columns

- Automatic float placement
  - into and across columns
  - maintaining relationships to call-outs
  - and produce pleasing (and compact) layout

- Grid typesetting
  - but use elements of varying and unknown heights

- Do everything automatically
  - but guess exceptions (without them being marked up)
Moving on . . .

1  The Ask

2  Some History

3  \TeX\’s Model

4  User Wishlist
   - Balancing & Floats
   - Columns
   - Column dependency
   - Rules & Backgrounds
   - At the border
   - Wishlist summary

5  Typesetting Challenges
   - Balancing
   - Right to left
   - Spaces & penalties

6  Summary
Brief history on multi-column attempts in \LaTeX

1982 \TeXbook\ describes code for index balancing

1988 multicol version 1.0a

1993 balancing by Patrick Daly

1999 \ltxgrid by Arthur Ogawa

1999-2000 Prototype of xor

2007 flowfram by Nicola Talbot

2011 adjmulticol by Boris Veytsman
Brief history on multi-column attempts in \LaTeX

1982  \TeXbook\ describes code for index balancing — specific use case only

1988  \texttt{multicol} version 1.0a — multiple columns, restriction on floats and footnotes

1993  balancing by Patrick Daly — manual adjustment needed

1999  \texttt{ltxgrid} by Arthur Ogawa — two-column, potential loss of material

1999-2000 Prototype of \texttt{xor} — general float handling, balancing, grids, but no production version

2007  \texttt{flowfram} by Nicola Talbot — general text blocks, no balancing, limited float support

2011  \texttt{adjmulticol} by Boris Veytsman — fixing some \texttt{multicol} limitations
Or in summary:

- 30 years have passed since TeX82
- Only a handful attempts
- No general solution

Why?
Moving on . . .

1. The Ask
2. Some History
3. \TeX’s Model
4. User Wishlist
   - Balancing & Floats
   - Columns
   - Column dependency
   - Rules & Backgrounds
   - At the border
   - Wishlist summary
5. Typesetting Challenges
   - Balancing
   - Right to left
   - Spaces & penalties
6. Summary
The \TeX\ model

We probably don’t have the extra computer resources for 100 people to typeset 40 or 50 page documents during the daytime.

\textit{Donald Knuth, Stanford 1981}

http://www.youtube.com/playlist?list=PL0AB2986EB62BB5FE
TEX was designed for documents with

- a single column
- left-to-right typesetting
- display mathematics (irregular baselines)
- offering paragraph optimization
The TeX model

Design basics

TeX was designed for documents with
- a single column
- left-to-right typesetting
- display mathematics (irregular baselines)
- offering paragraph optimization

The processing/typesetting is
- unidirectional: paragraph → galley → page
- with some limited multi-stream functionality (e.g., footnotes)
TEX was designed for documents with
- a single column
- left-to-right typesetting
- display mathematics (irregular baselines)
- offering paragraph optimization

The processing/typesetting is
- unidirectional: paragraph $\rightarrow$ galley $\rightarrow$ page
- with some limited multi-stream functionality (e.g., footnotes)

Paragraph shapes are
- specified by lines — not by vertical relationships
The \TeX{} model

Consequences

- Processing freezes the horizontal width
  - with no way to recover parts of an already typeset paragraph
  - instead, unset paragraph material needs to be kept
  - issues with reprocessing on macro level

Impermanent processing:
\begin{itemize}
\item \texttt{\LaTeX} \\
\item \texttt{\LaTeXe} \\
\end{itemize}

The \TeX{} model is the\textit{ fundamental factor}.

The balance between typesetting and layout is important.

\section*{Consequences}

- Processing freezes the horizontal width:
  - with no way to recover parts of an already typeset paragraph
  - instead, unset paragraph material needs to be kept
  - issues with reprocessing on macro level
The T\(\text{E}\)X model

Consequences

- Processing freezes the horizontal width
  - with no way to recover parts of an already typeset paragraph
  - instead, unset paragraph material needs to be kept
  - issues with reprocessing on macro level

- Size changes need to be modeled as paragraph shapes
  - conceptually wrong
  - precompiled paragraph shapes will give “incorrect results” if lines contain large objects
The \TeX model

Consequences

- Processing freezes the horizontal width
  - with no way to recover parts of an already typeset paragraph
  - instead, unset paragraph material needs to be kept
  - issues with reprocessing on macro level

- Size changes need to be modeled as paragraph shapes
  - conceptually wrong
  - precompiled paragraph shapes will give “incorrect results” if lines contain large objects

- No built-in concept for grids
  - the glue concept is effectively an orthogonal approach
The \TeX\ model

Consequences

- Processing freezes the horizontal width
  - with no way to recover parts of an already typeset paragraph
  - instead, unset paragraph material needs to be kept
  - issues with reprocessing on macro level

- Size changes need to be modeled as paragraph shapes
  - conceptually wrong
  - precompiled paragraph shapes will give “incorrect results” if lines contain large objects

- No builtin concept for grids
  - the glue concept is effectively an orthogonal approach

- Relationship between different streams is only maintained rudimentarily
  - loss of relationships inside the OR
Moving on . . .

1. The Ask
2. Some History
3. \TeX's Model
4. User Wishlist
   - Balancing & Floats
   - Columns
   - Column dependency
   - Rules & Backgrounds
   - At the border
   - Wishlist summary
5. Typesetting Challenges
   - Balancing
   - Right to left
   - Spaces & penalties
6. Summary
From the user wishlist . . .

Balancing or not . . .

- How do I fill the first column first?

- How do I avoid stretching in the last column?

- How do I prevent balancing of short texts?
From the user wishlist . . .

Balancing or not . . .

- **How do I fill the first column first?**
  — realized early on as `multicols*` environment

- **How do I avoid stretching in the last column?**
  — parameters, e.g., `\finalcolumnbadness` introduced

- **How do I prevent balancing of short texts?**
  — tricky, what are good options?
From the user wishlist . . .
Balancing or not . . .

- How do I fill the first column first?
  — realized early on as `multicols*` environment

- How do I avoid stretching in the last column?
  — parameters, e.g., `\finalcolumnbadness` introduced

- How do I prevent balancing of short texts?
  — tricky, what are good options?

  *Short text fragments do not balance nicely.*
  How should this material be distributed over the columns?  

  . . . or . . .

  *Short text fragments do not balance nicely.*
  How should this material be distributed over the columns?

  — manual solution with `unbalance`
  — parameter for minimal number of lines (to be added)
Can I have my floats automatically placed with `multicols`?

Can I have floats that span some of the columns?

Why are full width floats never placed on the current page?
From the user wishlist ...

Flotes

• Can I have my floats automatically placed with `multicols`?
  — not with the current approach used;
  balancing will throw off all placement calculations

• Can I have floats that span some of the columns?
  — not with the current approach used;
  — solutions could be (fairly easily) provided if
    • balancing is restricted
    • floats are manually placed by the user

• Why are full width floats never placed on the current page?
  — to avoid backtracking and recalculations;
  — these days this could be improved
Side note
Float placement costs

\begin{align*}
\text{brute force}(a, n) &= (a + 1)^n \quad \text{for } a \text{ areas and } n \text{ floats} \quad (1) \\
\text{trials}(a, n) &= \binom{n + a}{a} = \frac{(n + a)!}{n! \cdot a!} \quad (2)
\end{align*}

with \( a \geq c(c + 1) \) for \( c \) columns on a page.

An additional float waiting increases the trials by \( \frac{n + a + 1}{n+1} \)
Side note

Float placement costs

\[
\text{bruteforce}(a, n) = (a + 1)^n \quad \text{for } a \text{ areas and } n \text{ floats} \quad (1)
\]

\[
\text{trials}(a, n) = \binom{n + a}{a} = \frac{(n + a)!}{n!a!} \quad (2)
\]

with \( a \geq c(c + 1) \) for \( c \) columns on a page.

An additional float waiting increases the trials by \( \frac{n+a+1}{n+1} \)

- \( \text{trials}(8, 12) = 125.970 \)
- \( \text{trials}(10, 12) = 646.646 \)
- \( \text{trials}(6, 24) = 593.775 \)
- \( \text{trials}(8, 24) = 10.518.299 \)
- \( \text{trials}(3, 24) = 2.300 \)

This is the search space for one page, no global optimization!
From the user wishlist . . .

Columns

- Can I have columns of different width?
- Can I have different number of columns per page?
- Can I have arbitrarily shaped columns?

1. One two three four. 2. One two three four. 3. One two three four. 4. One two three four. 5. One two three four. 6. One two three four. 7. Red blue green yellow. 8. One two three four. 9. One two three four. 10. One two three four. 11. One two three four. 12. One two three four. 13. One two three four. 14. One two three four. 15. One two three four. 16. One two three four. 17. One two three four. 18. One two three four. 19. One two three four. 20. One two three four. 21. One two three four. 22. One two three four. 23. One two three four. 24. One two three four.
From the user wishlist . . .

Columns

- Can I have columns of different width?
  — not if the column content is automatically determined

- Can I have different number of columns per page?
  — only if the page break is manually chosen

- Can I have arbitrarily shaped columns?

  1. One two three four.  2. One two three four.  3. One two three four.  4. One two three four.  5. One two three four.  6. One two three four.
  7. Red blue green yellow.  8. four.  9. One two three four.  10. One two three four.
  11. One two three four.  12. One two three four.  13. One two three four.  14. One two three four.  15. One two three four.  16. One two three four.  17. One two three four.  18. One two three four.
  19. One two three four.  20. One two three four.  21. One two three four.
  22. One two three four.  23. One two three four.  24. One two three four.

  — not really in any generality
Reiterate . . .
Paragraph breaking

The situation:

- Paragraphs are broken into lines as a whole
- Formatted material cannot be reformatted (in general)
Reiterate . . .
Paragraph breaking

The situation:
- Paragraphs are broken into lines as a whole
- Formatted material cannot be reformatted (in general)

The consequences:
- For reformatting unformatted (macro level) material needs to be captured and used
- Any shape requirement (e.g., variation on horizontal width a page boundary) has to be anticipated beforehand and translated into a suitable \parshape
- However, TeX is not able to account for vertical variation, except by trial and error
From the user wishlist . . .
Columns (cont.)

- Can I force `multicols` to skip to the next column if there are only a few lines left?

- Can I have (fixed) content at the top of each column?

- Can I have variable content added at the top of each column, e.g., a continuation line in an index?

- Can I get my table broken across several columns/pages (with table headers repeated)?
From the user wishlist . . .

Columns (cont.)

- Can I force `multicols` to skip to the next column if there are only a few lines left?
  — yes, using the `\vfilneg` approach

- Can I have (fixed) content at the top of each column?
  — yes, fairly easily (no user interface provided)

- Can I have variable content added at the top of each column, e.g., a continuation line in an index?
  — yes, implemented for the `LaTeX Companion` index

- Can I get my table broken across several columns/pages (with table headers repeated)?
  — partly; without table headers `multicols` and `longtable` could be easily married
  — hard, with table headings or balancing requirements
From the user wishlist . . .
Act depending on current column

- Can I find out if some environment is typeset fully within a single column?
- Can I switch the line-numbering if the column changes?
- Can I produce pull-quotes sticking into the margin on outside columns?
- Can I force \texttt{multicols} to start the next column \textit{only} if in the first column, but not if in the second column?
Can I find out if some environment is typeset fully within a single column?

Can I switch the line-numbering if the column changes?

Can I produce pull-quotes sticking into the margin on outside columns?

Can I force `multicols` to start the next column *only* if in the first column, but not if in the second column?

— yes to all; it is possible to gather the necessary information with a two-pass algorithm and use it to provide typesetting variations
From the user wishlist . . .
Act depending on current column

- Can I find out if some environment is typeset fully within a single column?
- Can I switch the line-numbering if the column changes?
- Can I produce pull-quotes sticking into the margin on outside columns?
- Can I force `multicols` to start the next column *only* if in the first column, but not if in the second column?
  — yes to all; it is possible to gather the necessary information with a two-pass algorithm and use it to provide typesetting variations
  — if the typesetting significantly changes the column structure, more passes may be necessary
  — worst case scenario may be an impossible document, i.e., a document that alternates between two or more states
From the user wishlist ...

Rules

- Can I have the column rules colored?
- Can I have rules at the outside of `multicols`?
- Can I have a rule just between the 4th and 5th column?
- Can I (arbitrarily) customize the column rules?
From the user wishlist . . .

Rules

- Can I have the column rules colored?
  — yes, through \texttt{\columnseprulecolor}

- Can I have rules at the outside of \texttt{multicols}?

- Can I have a rule just between the 4\textsuperscript{th} and 5\textsuperscript{th} column?

- Can I (arbitrarily) customize the column rules?
  — yes, all this could be easily implemented

  — only question: what is an adequate user interface?
From the user wishlist . . .
Shades and backgrounds

- Can I color the background behind a `multicols` environment?

- Can I add borders?
From the user wishlist . . .

Shades and backgrounds

- Can I color the background behind a `multicols` environment?

- Can I add borders?

— yes to both; column boxes and positions are known, so it is easy to add background material

— no official interface though
From the user wishlist . . .
At the border

- Why is there no one-column `multicols` environment?

- How do I change the margins for a `multicols` environment?

- How can I interrupt `multicols` with single column material?

- How can I avoid the indentation after a `multicols` environment?
From the user wishlist ... 

At the border

- Why is there no one-column `multicols` environment?
  - simply because I didn't foresee the need (and it was simpler)
  - provided by `adjmulticol`

- How do I change the margins for a `multicols` environment?
  - provided by `adjmulticol`

- How can I interrupt `multicols` with single column material?
  - by stopping and restarting

- How can I avoid the indentation after a `multicols` environment?
  - manually, using `\noindent`
Wishlist summary . . .

Unresolvable wishes

- combination of conflicting requirements
Wishlist summary . . .

Unresolvable wishes

- combination of conflicting requirements

Hard problems

- touching \TeX\ processing / model limitations
  - may limit the applicable scenarios
  - may result in fragile “hacks”

- combination of requirements with layout interdependencies
  - multi-pass algorithms required
  - may result in impossible documents

- huge search space
  - no efficient algorithms
  - may result in sub-optimal solutions
Wishlist summary . . .

Unresolvable wishes

- combination of conflicting requirements

Hard problems

- touching \TeX\ processing / model limitations
  - may limit the applicable scenarios
  - may result in fragile “hacks”

- combination of requirements with layout interdependencies
  - multi-pass algorithms required
  - may result in impossible documents

- huge search space
  - no efficient algorithms
  - may result in sub-optimal solutions
Wishlist summary . . .

Unresolvable wishes

- combination of conflicting requirements

Hard problems

- touching \TeX processing / model limitations
  - may limit the applicable scenarios
  - may result in fragile “hacks”
- combination of requirements with layout interdependencies
  - multi-pass algorithms required
  - may result in impossible documents
- huge search space
  - no efficient algorithms
  - may result in sub-optimal solutions
Wishlist summary . . .

Unresolvable wishes

- combination of conflicting requirements

Hard problems

- touching Tex processing / model limitations
  - may limit the applicable scenarios
  - may result in fragile “hacks”
- combination of requirements with layout interdependencies
  - multi-pass algorithms required
  - may result in impossible documents
- huge search space
  - no efficient algorithms
  - may result in sub-optimal solutions

Soft problems

- missing (or bad) user interfaces
Moving on . . .

1. The Ask
2. Some History
3. \textsc{\LaTeX}'s Model
4. User Wishlist
   - Balancing & Floats
   - Columns
   - Column dependency
   - Rules & Backgrounds
   - At the border
   - Wishlist summary
5. Typesetting Challenges
   - Balancing
   - Right to left
   - Spaces & penalties
6. Summary
Sample typesetting challenges
Balancing considered harmful

The steps of the algorithm:

- Find a good starting column height
- Trial-cut all material into columns of that height
The steps of the algorithm:

- Find a good starting column height
- Trial-cut all material into columns of that height
- We fail the trial if
  - the badness of any column is higher than \texttt{columnbadness}
  - the last column is larger than first
  - a forced break ends up in the last column
  - previous trial height \textless natural height of first column \textless trial height
Sample typesetting challenges
Balancing considered harmful

The steps of the algorithm:

- Find a good starting column height
- Trial-cut all material into columns of that height
- We fail the trial if
  - the badness of any column is higher than columnbadness
  - the last column is larger than first
  - a forced break ends up in the last column
  - previous trial height < natural height of first column < trial height
- If the trail failed retry with a (slightly) larger column size
The steps of the algorithm:

- Find a good starting column height
- Trial-cut all material into columns of that height
- We fail the trial if
  - the badness of any column is higher than columnbadness
  - the last column is larger than first
  - a forced break ends up in the last column
  - previous trial height < natural height of first column < trial height
- If the trial failed retry with a (slightly) larger column size
- If the trial succeeded and the final column has a high badness (> finalcolumnbadness) set the column at its natural height and not stretched out
Balancing considered harmful

The issues

- Enlarging the column size may result in overflowing the available space
- The final result often shows compressed columns
Balancing considered harmful

The issues

- Enlarging the column size may result in overflowing the available space
  — resolved by introducing `\maxbalancingoverflow`

- The final result often shows compressed columns
Balancing considered harmful

The issues

- Enlarging the column size may result in overflowing the available space
  — resolved by introducing `\maxbalancingoverflow`

- The final result often shows compressed columns
Balancing considered harmful

The issues

- Enlarging the column size may result in overflowing the available space
  — resolved by introducing \maxbalancingoverflow

- The final result often shows compressed columns

  1. text 1 ________________ 4. text 4
  2. text 2 _________________
  3. text 3 _________________ 5. text 5
Balancing considered harmful
The issues

- Enlarging the column size may result in overflowing the available space
  — resolved by introducing \texttt{\maxbalancingoverflow}

- The final result often shows compressed columns

  1 text 1 __________________________ 4 text 4
  2 text 2 __________________________ 5 text 5
  3 text 3 __________________________

  — no (good) general solution possible
  — only alternative is to inspect the state of each column
    and use this to fail a trial
  \rightarrow danger: adequate solutions are dropped
Sample typesetting challenges

Typesetting right to left

\documentclass{article}
\usepackage[landscape,lines=30]{geometry}
\usepackage{multicol}
\usepackage{ptext} \usepackage{xepersian}
\begin{document}
\tableofcontents
\begin{multicols}{4}
\section{بخش اول [2-1]}
\begin{footnote} \texttt{p[ext][3]}
\section{بخش اول [2-1]}
\texttt{p[ext][4]}
\end{multicols}
\end{document}

Should be as easy as that ... and the task for multicol is simple, isn’t it?
Typesetting right to left

The result (wrong!)
Typesetting right to left
Corrected result

1 بخش اول
2 بخش اول
3 بخش اول

Frank Mittelbach

The stony road to complex page layout

The Ask
Some History
TeX's Model
User Wishlist
Balancing & Floats
Columns
Column dependency
Rules & Backgrounds
At the border
Wishlist summary

Typesetting Challenges
Balancing
Right to left
Spaces & penalties

Summary
Sample typesetting challenges
Space and penalty sequences

\begin{multicols}{2}
\begin{itemize}
\item \verb=\addpenalty= and \verb=\addvspace= logic in \LaTeX{} has a flaw
\item \texttt{multicol} makes this apparent
\end{itemize}
\begin{itemize}
\item \verb=\addpenalty= and \verb=\addvspace= logic in \LaTeX{} has a flaw
\item \texttt{multicol} makes this apparent
\end{itemize}
\end{multicols}
Space and penalty sequences

... and the result

- \addpenalty and \addvspace logic in \LaTeX has a flaw
- multicol makes this apparent

You see the issue?
Space and penalty sequences
... and the result

- \addpenalty and \addvspace logic in \LaTeX has a flaw
- \texttt{multicol} makes this apparent

You see the issue?
Right ... the columns don’t line up
Space and penalty sequences
So what’s the problem?

...\penalty -51 \%
1st \addpenalty
...\glue 8.0 plus 2.0 minus 4.0 \%
1st \addvspace
...\glue -8.0 plus -2.0 minus -4.0 \%
2nd \addpenalty
...\penalty -51 \%
\ dito
...\glue 8.0 plus 2.0 minus 4.0 \%
\ dito
...\glue(\parskip) 0.0 plus 1.0 \%
\ start of \item
...\hbox(6.83331+0.0)x345.0, glue set 323.88889fil
Space and penalty sequences
So what’s the problem?

\begin{itemize}
  \item LaTeX generates a sequence of \texttt{\addpenalty} and \texttt{\addvspace} calls
  \item The break is taken at the second penalty
  \item This leaves a zero space at the bottom of the column/page making the baseline shift downwards as \texttt{\prevdepth} is getting ignored
\end{itemize}
Moving on ...

1. The Ask
2. Some History
3. \TeX's Model
4. User Wishlist
   - Balancing & Floats
   - Columns
   - Column dependency
   - Rules & Backgrounds
   - At the border
   - Wishlist summary
5. Typesetting Challenges
   - Balancing
   - Right to left
   - Spaces & penalties
6. Summary
Summary & Conclusion

- If we take off the shoes, i.e., restrict the use case scenarios somewhat, then a lot can be achieved already within the limitations of \TeX{} (or Lua\TeX{})
If we take off the shoes, i.e., restrict the use case scenarios somewhat, then a lot can be achieved already within the limitations of \TeX{} (or \texttt{Lua\TeX{}})

Good automatic float placement is still a largely unresolved research area

— Not an engine question at this point!
Summary & Conclusion

- If we take off the shoes, i.e., restrict the use case scenarios somewhat, then a lot can be achieved already within the limitations of \TeX\ (or Lua\TeX)

- Good automatic float placement is still a largely unresolved research area
  — Not an engine question at this point!

- The paragraph shape model is defective
  — A typesetting engine needs to enable reformatting
  — Paragraph breaking algorithms should work on physical shape specifications not on line counts
Summary & Conclusion

- If we take off the shoes, i.e., restrict the use case scenarios somewhat, then a lot can be achieved already within the limitations of \TeX{} (or Lua\TeX{}).

- Good automatic float placement is still a largely unresolved research area
  — Not an engine question at this point!

- The paragraph shape model is defective
  — A typesetting engine needs to enable reformatting
  — Paragraph breaking algorithms should work on physical shape specifications not on line counts

- Multi-directional typesetting may need a better model
  — \TeX{}’s model is clearly defective, but Lua\TeX{}’s (Omega) model may need revisiting as well → looking forward to John’s talk
So is waiting necessary?
The stony road to complex page layout

Frank Mittelbach

The Ask
Some History
\LaTeX’s Model
User Wishlist
Balancing & Floats
Columns
Column dependency
Rules & Backgrounds
At the border
Wishlist summary

Typesetting Challenges
Balancing
Right to left
Spaces & penalties

Summary

Thank you

Or can it be helped a little?

Takayama 2009