High Quality Automatic Typesetting

Proposal for a new document model, typesetting language, and system architecture

Karel Skoupý

Computer Systems Institute
ETH Zürich, Switzerland
High Quality

- the printout should æsthetically please the (discriminating) reader
- result should be comparable to traditional methods
- traditional concern of $\LaTeX$ users
Automatic Processing

• content + design specification $\rightarrow$ visual document

• automatization of the whole preparation process

• ability to change the content and/or the design specification

• big gap between abstract specification and formatting control

• more abstract level of control $\rightarrow$ higher expressivity and productivity

• WYSIWYG works on a very concrete level

• TeX needs a lot of concrete manual tuning for complex layouts
Implementation Limitations of \LaTeX

- Great difficulty making extensions
  - 20 years old implementation
  - monolithic code
  - obscure dependencies
  - overkill optimizations
  - lack of abstraction

- Poor integrability and reusability
  - no modularization
  - rigid program interface
The $\mathcal{NTS}$ Project

- Complete modular redesign of $\LaTeX$
- Reimplementation in Java
- Strict functional identity (compatibility) with $\LaTeX$
- Extendibility and reusability
Figure 1  Hierarchy of the $\mathcal{N}_T S$ packages
Figure 2  $NITS$ in its Full Speed
Conceptual Limitations

- Rectangular document model
- Static processing semantics
- Low level input/control language
- Monolithic system architecture
The Buddha told Ananda, “All the aspects of everything in the world, such as big and small, inside and outside, amount to the dust before you. Do not say the seeing stretches and shrinks. Consider the example of a square container in which a square of emptiness is seen. I ask you further: is the square emptiness that is seen in the square container a fixed square shape, or is it not fixed as a square shape?

If it is a fixed square shape, when it is switched to a round container the emptiness would not be round. If it is not a fixed shape, then when it is in the square container, it should not be a square-shaped emptiness. You say, ‘You do not know where the meaning lies.’ The nature of the meaning being thus, how can you speak of its location?

Ananda, if you wished there to be neither squareness nor roundness, you would only need to remove the container. The essential emptiness has no shape, and so do not say that you would also have to remove the shape from the emptiness.

If, as you suggest, your seeing shrinks and becomes small when you enter a room, then when you look up at the sun shouldn’t your seeing be pulled out until it reaches the sun’s surface? If walls and eaves can press in and cut off your seeing, then why if you were to drill a small hole, wouldn’t there be evidence of the seeing reconnecting? And so that idea is not feasible.”

Figure 3  General shape frames in Adobe InDesign
Document Model

- Only rectangular shapes – any graphics is just a box
- Very poor means for specification of non-rectangular text blocks
- Insufficient means for texts along curves
Various \LaTeX\ drawing languages can naturally incorporate texts into drawings because those languages are built on top of \LaTeX. Some of them support also scaling and rotating of textual objects. The most fancy features are provided by the \texttt{PStricks} package. It is possible to bend a line or even a text block (with proper kerning) along a general curve. However, it is not possible to influence back the \texttt{FiG}-matting by the curve’s shape. \[ \sum_{i=0}^{n} a_i \] The information exchange is unidirectional in this case; everything is prepared in advance and passed to DVI specials and the rendering is postponed until printing. In \texttt{METAPOST}, there are two ways to work with text. The first is to use the \texttt{infont} operator which makes a picture of a character string using a given PostScript font. However, the glyphs are just put next to each other and without kerning. The second is to call \TeX\ to perform arbitrary typesetting tasks and the result is then accessible as a fixed picture with a known bounding box. The two ways just described can be even combined making it possible to use individual character glyphs and to let \TeX\ determine the kerning dimensions (one by one of course). Having collected this information, the \texttt{METAPOST} code can typeset the properly kerned text along a curve. This trick is used in MetaFun and nicely illustrates that people can do just everything with \TeX\ and \texttt{METAPOST}. Perhaps someone will write \texttt{METAPOST} macros which implement optimal paragraph breaking.

**Figure 4** Non-robust \LaTeX\ solution for wrapping text
More General Document Model

- introducing paths and graphical facilities (from METAPOST)
- unification of text, font, and graphics objects
- uniform representation of (composite) objects
- natural blending of text and graphics
The Buddha told Ananda, "All the aspects of everything in the world, such as big and small, inside and outside, amount to the dust before you. Do not say the seeing stretches and shrinks. Consider the example of a square container in which a square of emptiness is seen. I ask you further: is the square emptiness that is seen in the square container a fixed square shape, or is it not fixed as a square shape? If it is a fixed square shape, when it is switched to a round container the emptiness would not be round. If it is not a fixed shape, then when it is in the square container it should not be a square-shaped emptiness. You say you do not know where the meaning lies. The nature of the meaning being thus, how can you speak of its location? Ananda, if you wished there to be neither squareness nor roundness, you would only need to remove the container. The essential emptiness has no shape, and so do not say that you would also have to remove the shape from the emptiness. If, as you suggest, your seeing shrinks and becomes small when you enter a room, then when you look up at the sun wouldn’t your seeing be pulled out until it reaches the sun’s surface? If walls and eaves can press in and cut off your seeing, then why if you were to drill a small hole, wouldn’t there be evidence of the seeing reconnecting? And so that idea is not feasible."

Figure 5  Paragraph shape defined by an orthogonal polygon
Example of `\textit{blockshape} No. 2`

The Buddha told Ananda, "All the aspects of everything in the world, such as big and small, inside and outside, amount to the dust before you. Do not say the seeing stretches and shrinks. Consider the example of a square container in which a square of emptiness is seen. I ask you further: is the square emptiness that is seen in the square container a fixed square shape, or is it not fixed as a square shape? If it is a fixed square shape, when it is switched to a round container the emptiness would not be round. If it is not a fixed shape then when it is in the square container it should not be a square-shaped emptiness. You say you do not know where the meaning lies. The nature of the meaning being thus, how can you speak of its location? Ananda, if you wished there to be neither squareness nor roundness, you would only need to remove the container. The essential emptiness has no shape, and so do not say that you would also have to remove the shape from the emptiness. If, as you suggest, your seeing shrinks and becomes small when you enter a room, then when you look up at the sun shouldn’t your seeing be pulled out until it reaches the sun’s surface? If walls and eaves can press in and cut off your seeing, then why if you were to drill a small hole, wouldn’t there be evidence of the seeing reconnecting? And so that idea is not feasible."

Figure 6  As before but with irregular line heights
Representation and Processing Semantics

- Interdependent input and formatting
- Rigid representation of paragraphs and pages
  - paragraphs are formatted first and never reformatted
  - simple page breaking cannot influence formatting of sub-objects
  - big obstacle for more sophisticated page breaking
  - and for more complex page and column layouts
- broader context optimization limited to paragraphs only
Dynamic Representation and Processing

- separation of input (representation building) and formatting (representation transformation)
- keeping broader dynamic context (whole chapters, documents) allowing global optimization
- information richness of representation
- self-adaptable objects with definable behavior
Input and Control Language

- Primitive macro language with obscure and context dependent syntax and syntactic rules
- Primitive and non-extensible type system
- No provision for modularity
- Unclean specification not separated from the implementation (over-specified)
- \TeX\ macro-language is powerful for input manipulation but not for object representation (box) manipulation
- Incomplete set of primitives (missing \texttt{\textbackslash last*}, \texttt{\textbackslash un*})
Proposal for Languages

- clean syntax and semantics (with formal specification)
- proper and extendible (definable) type system
- modular: separation of interface and (exchangeable) implementation
- complete: full elegant programmer control, no need for dirty tricks
- regular and basic: providing primitives, not solutions
- open: user constructs as powerful and convenient as primitives
- universal API for different language bindings
• possibly different languages for:
  – input (pre)processing
  – layout specification
  – object representation manipulation
System Architecture

- \TeX{} architecture is monolithic and not extendible
- \TeX{} external API non flexible (input, log, output)
Proposed Architecture

- pluggable frontends: \( \TeX \), XML, \ldots
- pluggable backends: PS, PDF, DVI, plain text content, \ldots
- pluggable alternative algorithms and policies
- various language bindings: scheme, python, \ldots
- reusable subsystems (modules)
Conclusion

- T\(\text{E}\)X is already (almost) perfect for tasks it was designed for (and many more), no need to improve

- more general document and processing model needed for more challenging tasks

- higher level modular input and control language can increase expressivity and productivity

- open modular architecture can improve flexibility and applicability

- better framework for document/interface design

- it should be easier to express the document design and let the machine to put it into effect
Related Links

• $\mathcal{N}_T S$ code: ftp://dante.ctan.org/pub/tex/systems/nts/

• My papers about $\mathcal{N}_T S$: http://www.inf.ethz.ch/~skoupy/papers/