Towards LaTeX 3.0

Frank Mittelbach
Electronic Data Systems (Deutschland) GmbH
Eisenstraße 56 (N15)
D-6090 Rüsselsheim
Federal Republic of Germany
Bitnet: `pzf5hz@ruipc1e`

Rainer Schöpf
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Str. 10
D-1000 Berlin 31
Federal Republic of Germany

Abstract

LaTeX is a very valuable tool for document composition. As a TeX macro package, it is unique in its concept of logical commands, at the same time retaining enough flexibility with visually oriented commands to allow the user a relatively easy correction of an automatically chosen layout. This fact makes it far superior to the plain and $\mathcal{AMS}$-TeX macro packages when it comes to professional applications.

Therefore the LaTeX re-implementation project is certainly one of the most important efforts to "expand TeX's horizon". This is the place to link TeX with the modern developments like SGML. This paper describes the current status of the re-implementation of LaTeX.

1 Objectives of the LaTeX project

LaTeX [6] was developed to serve the specific needs and designs found in documents of natural science, whereas the needs of other fields are more or less neglected. Many layouts and concepts cannot be realized in the present LaTeX, and even those that can be realized are often falsely flagged as "impossible in LaTeX".

One of the main objectives of the LaTeX project is, therefore, to redesign the style file interface by incorporating a broader spectrum of possibilities. At the same time, we try to structure this interface in such a way that it satisfies the needs of a designer. This means that desired layouts should be specifiable preferably through parameters and generic functions that allow a wide range of varieties.

An ensuing objective is the proper documentation of the new interface. It should guide the designer in the evaluation of a new layout, allowing him to use the full power of the interface within a short period of time.

As a third objective, we feel it necessary to reevaluate LaTeX's internal concepts and reverse those that have been proven inadequate.

In our talks at last year's conferences [8, 9, 10, 11], we presented a concept for a re-implementation of LaTeX. After discussing this topic, Leslie Lamport and one of the authors (FMi) agreed on a two-step procedure, first redesigning the style interface, and then enhancing the user interface.[1]

Further discussion throughout this year has shown that this plan is not feasible as the internal style file interface is affected too greatly by enhancements in the user interface, and vice versa. Therefore, we abandoned this idea and decided to merge both steps, at least temporarily.[2] As a result, the discussion then focused on three different major topics, the enhancements and changes to the user inter-

1. This was published as the update section in [8] and [7].
2. We feel that it is still sensible to defer certain parts of the revision to a later stage. However, this will only concern internals of the implementation and not induce any changes to the user or the style-designer interface.

face, the revision of the style file interface, and the re-evaluation of internal concepts.

In the following sections, we will discuss the topics which we have been concerned with since last year's conference and the state of their realisation.

2 The User Interface

Any change to the user interface of an existing program always opens the question of compatibility. While we judge this question as very important, we feel that it is not justified, for the sake of upward-compatibility, to leave all existing features untouched, even when they have proven to be inadequate. This means that we try to keep these sorts of changes small, and devise a possibility to emulate LaTeX 2.09 in the new LaTeX to allow processing of older documents with few or no changes.

2.1 Attribute concept. It has been generally agreed in the ongoing discussion that an attribute concept as supported by DCF GML [4, 5] and SGML [3] would be a great improvement. Since this is a major change, it was discussed whether such a concept would render the optional arguments obsolete. But as the number of LaTeX users is far greater than most people think, such an incompatible change in the syntax is not feasible. In addition, the old optional arguments and star forms provide convenient short forms for the most important attributes. While there have been several proposals for an attribute syntax as well as one prototype implementation, a final decision has not been made as yet. But it seems probable that this concept will only be available for environments, not for ordinary commands. We think that this will be sufficient since it is planned to provide environment forms of all text producing commands (cf. 2.7).

2.2 Robust and fragile commands. The distinction between robust and fragile commands will no longer be present. Instead, all text in moving arguments will be automatically protected against expansion.[3]

To allow the style file writer the specification of text that has to be expanded, there will be a command that removes or partly removes this protection.

We do not consider it necessary to give this feature to the user. Hence, this is not available in the user interface.[4]

2.3 Font selection. The new font selection scheme is already being distributed as beta test version for LaTeX 2.09 and seems to be working very well. It is also the basis for the `amsfonts` style option that makes the AMSFONTS collection available for LaTeX and is part of the AMS-LaTeX distribution [1].

Nevertheless, the current implementation should not yet be regarded as the final product. Time and users' demands will show whether it has to be improved.

2.4 Front matter. The specification of preliminary material is one of the parts which are not handled properly in LaTeX 2.09. Although this topic has not been discussed in depth so far, this might be one of the places where the syntax of the new LaTeX might differ in an incompatible way.

2.5 Tables. The extensions of the `array` and `tabular` environments by Frank Mittelbach [12] seem to be widely accepted. Several further extensions are conceivable, but this needs careful evaluation. There is also a new implementation of these environments by Denys Duchier that includes the extended syntax. This implementation looks very promising and can probably serve a basis for table handling in the new LaTeX.

We will provide a command to specify notes to tables working similar to the `\footnote` command inside a `minipage` environment.[5]

2.6 Math. The `amstex` style option for LaTeX 2.09 implements most of the features of AMS-TeX in LaTeX syntax (such as `\begin{align}...\end{align}` instead of `\align...\endalign`). As this is now being distributed by the AMS, users are able to typeset complicated math formulas in LaTeX without falling back to plain TeX's idiosyncrasies [1].

However, the implementation still has some loopholes that need to be eliminated in future versions.[6]

2.7 Text producing arguments. All commands with arguments in which the user specifies

3. The approach of LaTeX 2.09 to expand everything by default is counter-intuitive and a common source of nasty errors.

4. Expansion is normally necessary to cope with problems presented by the asynchronous output routine mechanism together with macros that change their contents, so that it is essential to write the expansion and not the macro name to a file, etc.

5. Using `\footnotemark` and `\footnotetext` commands inside a table that (additionally) has to be put inside a `minipage` environment is another counter-intuitive concept of LaTeX 2.09.

6. Some features do not work correctly in boundary cases. This is partly due to limitations in the current LaTeX, e.g., the primitive handling of `\begin...\end` (see below), etc.

text to be typeset (e.g., `\fbox`) will also be available in an environment form to allow their use in user-defined environments.

2.8 Verbatim input. The use of the `\verb` command will be possible in all circumstances.[7] A similar extension of the `verbatim` environment is not possible.[8] But this restriction is lessened by the possibility to use the environment form of the respective command in which the `verbatim` environment may be used.

2.9 Float positioning. LaTeX 2.09 was designed for documents containing only relatively few floats.[9] The new implementation will improve the float position algorithm and the user's control. This might include some sort of an 'h' option that really means "here".[10] There have also been proposals to prohibit the use of multiple captions within one float, thus allowing the document style to position the caption according to its own rules. But these topics have not yet been discussed thoroughly enough to present a final concept.

2.10 Bibliographies. The handling of citations and bibliographies will probably change to support several conventions. Since this topic depends on the development of the new BibTeX to some extent, it is not yet clear what is actually implementable.

Specific problems concerning citations and the interaction with BibTeX are discussed in [14] and in [13].

2.11 Omitting environment end tags. The implementation of a prototype for error recovery in case of unmatched `\end` tags 4.1 has shown that it is possible to implement the concept of implied `\end` tags. This feature will help in writing SGML parsers that produce LaTeX output. Whether the detection of implied `\begin` tags, e.g., the omission of the first `\item` in a list environment, can be easily implemented requires further testing.

3 The Style-Designer Interface

As we stated above, the main goal for the design of the style file interface is making it easily applicable. Therefore, the new interface will contain a lot more generic commands allowing for the specification of a wide range of layouts with a minimum of effort.

3.1 International language support. Support for more than one language (US English) was one of the key issues that triggered this LaTeX re-implementation project. In the new implementation, all textual representations in style files will be settable. While this is certainly not sufficient to support the different typographic conventions of different countries, it allows for typesetting foreign texts within the usual typographic conventions.

3.2 Hooks. For the implementation of certain layouts, it is often necessary to carry out specific actions at well-defined points, e.g., the footnote placement algorithm of this article has to be initialized at `\begin{document}`. For this type of application, many of the internal commands will contain hooks that allow the style file writer to add code to these commands without overwriting the original definitions.[11]

3.3 Generic section headers. One goal for the style file interface is to provide the designer with a generic heading macro which implements a broad range of layouts by varying certain parameters. It is clear that a proper balance has to be found between the internal complexity of such a command and the number of different layouts which are specifiable through it. Several different syntax proposals have been discussed so far, but the discussion hasn't reached a satisfactory conclusion, as yet.

The new mechanism will probably provide a specification for headings, in which the designer has complete control over heading layout, e.g., positioning supplied text[12], ornaments[13] and the like. It is planned to support the following general types of headings:

- Vertically oriented headings, where the heading is separated by white space from preceeding and following text.[14] There will be parameters to allow the heading to extend into one or both margins.

7. However, due to limitations of the TeX program itself, the use of multiple blanks in one `\verb` command will not be supported in all cases.

8. In LaTeX 2.09 neither the `\verb` command nor the `verbatim` environment may be used in arguments.

9. If, for example, the space for floats is larger than the surrounding main text, as is often the case in appendices of manuals, etc, LaTeX 2.09 is seldom able to compile the document without running out of memory space, even if the float parameters are given full flexibility.

10. We are aware of the problem of handling previously deferred floats of the same kind.

11. In LaTeX 2.09 a lot of style options are incompatible with each other, simply because they redefine the same internal macro.

12. For example, 'Chapter' in the `\chapter` command.

13. Rules and dingbats, etc.

14. This layout has already been realized to a certain extent in the `\@startsection` command of LaTeX 2.09. But this generic macro is not able to specify the layout of the heading (except setting the used font), so that it can not even be used for standard headings like the `\chapter` command.

- Horizontally oriented headings, where the heading is partially or fully placed into one of the margins beside the following text.[15] A critical problem with this sort of layout is to guarantee that enough space is available and that any necessary hanging indentation (for headings placed only partly into the margin) is applied up to the necessary point.
- Run-in headings, where the text following continues on the same line as the heading.[16]

The designer will be given tools for specifying the heading layout in an easy manner, so that it is possible to vary the layout depending on things like the length of the heading text, the presence or absence of a heading number, etc.

It is planned to allow for the specification of a minimal amount of text that has to follow a heading.[17]

3.4 Generic table-of-contents entries. What has been said in the last subsection about generic section headers applies to the formatting of entries in the table of contents as well. This has not yet been discussed in detail, but we hope that a proper implementation of generic section headers can be used as a starting point for generic toc entry formatting. The same mechanism can then be used for other types of tables as well (`lof`, `lot`, etc.).

3.5 Generic lists. Necessary extensions and corrections of the generic list environment have been already discussed in [8, 11]. There have been a few proposals concerning lists, but this topic needs further attention, as it affects a major part of most style files.

3.6 Parameter tables. There has been a proposal to group certain parameters into tables to make their structure and dependence visible. One item that will probably change on the style designer level in this way is the specification of default parameters for different levels of lists. But it is not yet clear, whether such a concept is implementable in an efficient manner.

3.7 Paragraph design. The specification of paragraph layout (such as different forms of ragged right typesetting) will be improved. The designer will be given the possibility to specify such layouts not only for the main text, but also for footnotes, floats, etc.[18]

3.8 Toc levels. There are book designs which require several tables of contents, e.g., one for the whole book and those referring to each chapter. This touches on the topic of auxiliary file handling (cf. 4.2) since the current mechanism does not allow more than one table per document. There will be support to select only certain entries of a tàble of contents for printing. A prototype implementation for this feature was written by Nico Poppelier.

3.9 Documentation. The interface between the style files and the LaTeX kernel will be documented properly. We will provide a complete description as well as a number of examples.

4 Internals

4.1 Error recovery. In the current LaTeX, an omitted or misspelled environment name usually produced a lot of error messages and often suppressed any further compilation of the document. In the new implementation, certain classes of environment errors are detected and corrected without damaging the output. For this complex, a prototype implementation is currently undergoing alpha testing. It implements the following features:

Incorrect `\begin` tag If the user misspells the name of the environment desired inside the `\begin` tag, an error message is generated and the offending `\begin` tag is ignored.[19] However, the user is allowed to insert the correct environment name by specifying `i\begin{`⟨*envir*⟩`}` in response to the error message.[20]

Incorrect `\end` tag When an incorrect `\end` tag is encountered, e.g., `\begin{bar}` ... `\end{foo}`, the new LaTeX tries the following recovery:

1. If `\end{foo}` is unknown, we assume that the user misspelled the name and recover by replacing `\end{foo}` with `\end{`⟨*curenvir*⟩`}`. This will produce an error message but will allow safe continuation of the compilation afterwards.
2. If `\end{foo}` is a legitimate `\end` tag, i.e., if the corresponding internal environment start com-

15. This sort of layout is not supported in the current version.

16. Again, this layout is provided in LaTeX 2.09 but does not allow specifying the layout of the heading, e.g., punctuation marks at the end, underlining, etc.

17. In LaTeX 2.09 this is the fixed amount of two lines of text. It might be possible that a more general implementation will fail in special cases due to TeX limitations.

18. In LaTeX 2.09, this is not possible without redefining several internal macros, because the paragraph shape parameters are reset at several points to fixed defaults.

19. Of course, this mechanism will be triggered only if the user did not exchange one environment name for another.

20. In LaTeX 2.09, this response would result in TeX error messages at the end of the compilation.

mand is defined, we check to see whether there are any unresolved `\begin{foo}` environments.

(a) If so, the currently open environment is closed by inserting an `\end{`⟨*curenvir*⟩`}` tag.[21] Afterwards `\end{foo}` is tried again. This mechanism will close all open environments until the correct one is found. Depending on the status of an internal variable, this will either produce an error message, or a warning message, or will be executed silently to allow for the implementation of implied `\end` tags.

(b) If there is no open `\begin{foo}` we simply ignore the `\end` tag after issuing an appropriate error message. The underlying idea is that this `\end` tag is probably left over after moving some text in the source around.

4.2 Auxiliary file handling. We will implement a two-step approach for `.aux` files where all information is written to an intermediate file which is then copied to the 'real' `.aux` file at the end of the run. As a result, there will be only one `.aux` file instead of the many files produced by LaTeX 2.09 when the `\include` command is used. The new scheme will make it possible to preserve cross-reference information if a compile run ended prematurely.[22] In addition, it will be possible to detect whether `\include` files have to be re-compiled because of changes in other parts of the document, or not.

4.3 References. The use of symbolic references will be extended to include textual references, if desired.[23] Whether this will result in some changes concerning the user interfaces not has not been discussed so far. If it is feasible, we will also provide the possibility for hierarchical references.[24]

4.4 Page selection. To provide easy access to the different parts of a document at the printing level, we plan to record certain document structure information in the `\count` registers 0–9. Besides the usual page counter in `\count` 0, this might include the physical page number, the current chapter, section, or subsection, ... number, to allow the printing of, e.g., Chapter 6 or "all preliminary pages"[25] by giving a simple page selection pattern to the printer driver.[26]

4.5 Plain TeX compatibility. We tend to build up the new LaTeX from scratch, i.e, not to read in the plain TeX format (or a nearly identical variant) as a basis when building a format file.[27] This does not mean the useful functions, `\mathchar` definitions, etc. of `plain.tex` will be discarded, but concepts which are obsolete in the LaTeX environment or macros that can be implemented in a better way will be replaced or removed.

5 Beta testing

It took LaTeX about three years to develop into a stable system. To avoid needing a similar period of time when switching to the new version, we plan to run the new version throughout the development at a few selected sites for beta testing. So, if you are a maintainer of a TeX installation and think that you can persuade your users to play willing (or unwilling) guinea pigs, please, contact one of the authors. Your installation should have the following characteristics:

- A running TeX 3.0 preferably (but not necessary) with drivers supporting the virtual fonts introduced with TeX 3.0.
- A fair amount of LaTeX processing in a fully supported LaTeX 2.09 environment (including a customized Local Guide) and at least one user with some experience in style writing.
- A working eMail connection.
- A maintainer (you) who is willing to
  - provide backups and fast user support in case something goes wrong
  - send in bug reports, if necessary
  - compile new formats when updates arrive.

21. The implementation of this part of the recovery is not as straight forward as it may seem. The term ⟨*curenvir*⟩ does not necessarily refer to the innermost open environment because it may be possible that a user defined environment calls other environments in its body. In such a case, the calling environment has to be closed first, because the inner environments are resolved in its `\end` tag.

22. The current implementation writes directly to the `.aux` file, so that its contents are damaged or lost when an error occurs.

23. In the current LaTeX, references can only be made to counter values, or more exactly, to a combination of counter values. This will lead to problems if document styles decide to use unnumbered headings, for example.

24. In LaTeX 2.09, this has been realized for the `enumerate` environment, to some extent.

25. In LaTeX 2.09, it is often not possible to print a specific set of pages that have the same TeX page numbers as other pages that come earlier in the document.

26. With most printer drivers, the first page to print can be selected by specifying a pattern to be matched against TeX's `\count` registers 0–9. We propose that drivers also allow for the specification of a pattern for the pages to be included.

27. Currently, LaTeX is built on top of `lplain.tex` which differs from `plain.tex` in only a few places. This means that even the funny keyboard support (for SAIL terminals) is included that will give you in certain situations a '⊕' character if you key in `$^^M$`.

The work that has to be carried out by the beta testers should not be underestimated. It is probably a time-consuming commitment since a lot of organizing is usually involved. Nevertheless, we hope that there will be enough people around willing to help us in this stage of development, so that we can finally return a product to the user that will have the same success as the current LaTeX had.

Throughout the beta testing phase, LaTeX will have to show its abilities in real life situations. As TeX and LaTeX are currently finding their way into new areas, we hope that people from these fields will take the opportunity to test whether the new LaTeX matches their needs by participating in the testing phase. This is the time when it is still possible to correct errors before they become established as unfortunate facts.

References

[1] American Mathematical Society, Providence. *AMS-LaTeX Version 1.0 User's Guide*, July 1990.

[2] Brüggemann-Klein, Anne, editor. *1989 EuroTeX Conference Proceedings*, 1990. To appear.

[3] Bryan, Martin. *SGML: an author's guide to the standard generalized markup language.* Addison-Wesley, Woking, England; Reading Massachusetts, second edition, 1988.

[4] *DocumentComposition Facility: Generalized Markup Language Starter Set User's Guide.* New York, fifth edition, May 1987.

[5] *Composing Documents with the Generalized Markup Language.* International Business Machines Corporation, New York, second edition, March 1988.

[6] Lamport, Leslie. *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, Massachusetts, 1986.

[7] Mittelbach, Frank and Rainer Schöpf. "A new font selection scheme for TeX macro packages — the basic macros." *TUGboat*, 10(2):222 – 238, July 1989.

[8] Mittelbach, Frank and Rainer Schöpf. "With LaTeX into the nineties." In Thiele, Christina, editor, *1989 Conference Proceedings*, volume 10#4 of *TUGboat*, pages 681 – 690. TeX Users Group, December 1989.

[9] Mittelbach, Frank and Rainer Schöpf. "LaTeX dans les années 90." *Cahiers GUTenberg*, (6):2 – 14, July 1990. French translation of [11] and of some parts of [10].

[10] Mittelbach, Frank and Rainer Schöpf. "LaTeX limitations and how to get around them." In *1989 EuroTeX Conference Proceedings*, Anne Brüggemann-Klein [2].

[11] Mittelbach, Frank and Rainer Schöpf. "With LaTeX into the nineties." In *1989 EuroTeX Conference Proceedings*, Anne Brüggemann-Klein [2].

[12] Mittelbach, Frank. "A new implementation of the array – and tabular – environments." *TUGboat*, 9(3):298 – 314, December 1988.

[13] Read, David. "Towards BIBTeX style-files that implement principal standards." *TeXline*, (10):2 – 8, May 1900.

[14] Wonneberger, Reinhard and Frank Mittelbach. "BIBTeX reconsidered." In Guenther, Mary, editor, *TeX 90 Conference Proceedings*, volume 12#1 of *TUGboat*, pages 111 – 124. TeX Users Group, March 1991.