StarTEX — a TEX for beginners

Dag Langmyhr
Department of Informatics
University of Oslo
dag@ifi.uio.no

1 Background

In many courses taught at the Department of Informatics at the University of Oslo, students are required to write a short report or essay, typically 3–8 pages. A suitable text processing tool for this must fulfill the following requirements:

- It must run under UNIX, as all our student computers use UNIX. That excludes *Scientific Word* as well as *Word* and similar programs.
- It must not be too expensive. That excludes *FrameMaker*.
- It must provide quality mathematical typesetting, as several courses involve substantial amounts of mathematics. That excludes *Word Perfect* and *Eqn+Troff*.
- It should be easy to learn and use, yet robust. This is a point against *LATEX*.

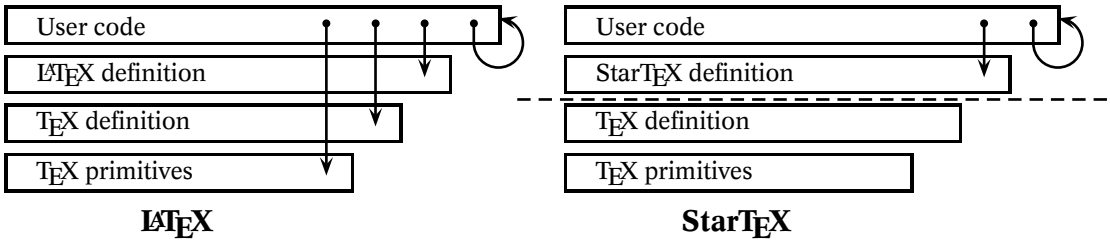
My conclusion is that the ideal text processing tool for this group of users does not yet exist. It ought, however, to be possible to make a better tool for them, and the result of this project is StarTEX ("a starter's TEX").

1.1 Choosing TEX as base

To achieve the desired quality in mathematics, it was decided to develop StarTEX as a TEX extension, making it a kind of "simple cousin" of LATEX's.

The text processing program TEX was created by Donald Knuth[1] in 1982. It is recognized as probably the best available tool for mathematical typesetting today, and it is widely used in academic institutions. It runs on all commonly used computers, and it is available free.

Figure 1: The implementation levels of LaTeX and StarTeX



TeX provides only a low-level set of commands, but it has a macro language for building extensions; the best known such extensions are LaTeX[2] and $\mathcal{AMS}$-TeX[3].

2 Designing StarTeX

StarTeX was designed to be an improvement on LaTeX for one particular group of users: students writing their first report.

- StarTeX is not for general use. This implies that the set of commands should be chosen with our users in mind. In particular, StarTeX should have fewer and simpler commands than LaTeX, and these commands should be more specialized for our use.
- StarTeX should provide a notation that causes the user fewer problems.
- StarTeX should be more robust than LaTeX.
- StarTeX should provide better error messages and error recovery than LaTeX.

2.1 Abstraction levels

Even though LaTeX is a complete package implemented on top of TeX, it does not provide a separate abstraction level, as there is no separation between the various implementation levels. The users can call their own macros as well as LaTeX macros, plain TeX macros and TeX primitives, as shown in figure 1.

This lack of separation means that the LaTeX users have access to several hundred commands unknown to them. This can lead to various kinds of problems; for example, a student defined a macro for her name:

```
\def \else {Else Hansen}
```

This unintentional redefinition of a TEX primitive resulted in utter chaos.

StarTEX makes it impossible for its users to call TEX macros and internals.

2.2 Command notation

StarTEX uses a different command notation than LATEX: <command> rather than \command. This notation — originally proposed in the TEX world by Philip Taylor[4] — has the following advantages:

- There is only one special character ('<') rather than ten ('\', '{', '}', '#', '$', '%', '^', '_', '~' and '&').
- Spaces following the command are no longer a problem. (LATEX generally ignores spaces following a command, but not always.)
- It is possible to enforce complete separation between the various implementation levels.
- It is easy to check every command when it is called, thus enabling better error detection.
- The commands may now contain almost any character, not just letters.
- The commands can be insensitive to casing, making <startex> and <STARtex> and even <starTeX> variant forms of the same command.
- The notation is the same as in HTML with which some students are familiar.

2.2.1 Environments

LATEX uses three different notations for environments:

{\bf *text*} and \textbf{*text*} and \begin{bf}*text*\end{bf}

This abundance of notation is the cause of many student problems:

- It is difficult to remember which one to use. For instance, if you write

  ```
  \abstract{This is ...}
  ```

  rather than the correct

```
\begin{abstract}
  This is ...
\end{abstract}
```

no errors will be reported, but the whole document will be set with a smaller font.

- It is difficult to match the environment initiator and terminator. If you get a message about "\end occurred at level 1," it is not obvious how to detect the unmatched { or \begin causing the problem.
- It is more difficult for LaTeX to correct errors.

StarTeX solves these problems by using the same notation as HTML for environments: <title>. . . </title>.

2.3 Specialized commands

LaTeX has an impressive graphics package for inserting illustrations in various kinds of formats. The user may also control such insertion parameters as scaling, rotation, and whether the illustration may float to a nearby page. All this is necessary when writing complex documents like a book, but it complicates life for the novice users. For instance, just to include a simple POSTSCRIPT illustration, something like the following code is necessary:

```
\usepackage[dvips]{epsfig}
    :
\begin{figure}
  \caption{Caption text}
  \begin{center}
    \epsfig{file=filename.eps,...}
  \end{center}
\end{figure}
```

Even though it is straightforward, there are two environments and two commands involved.

StarTeX is more specialized in its notation:

- Since all out printers use POSTSCRIPT, the illustrations must be in that format, so there is no need to specify it.
- All illustrations are regarded as floating material.
- The illustrations are automatically scaled to a suitable size (80% of the text width or 40% of the text height, whichever is the smaller).

Table 1: A small table sample

Index	Data
12	199
17	0

This makes it possible to define a simpler notation (at the cost of less user control):

```
<psfig>[filename.eps]Caption text
</psfig>
```

2.4 Robust notation

As mentioned in the preceding paragraph, the LaTeX notation is reasonably straightforward, but it is not very robust. If you for instance want a simple table like the one shown as table 1, the LaTeX code would be:

```
\begin{table}
  \caption{A small table sample}
  \begin{center}
    \begin{tabular}{|c|c|} \hline
      \textbf{Index}& \textbf{Data}\\ \hline
      12& 199\\ \hline
      17& 0\\ \hline
    \end{tabular}
  \end{center}
\end{table}
```

If you make a small mistake and forget a \\, you will get a long burst of error messages, none of which will indicate the real cause of the trouble.

When defining StarTeX, I tried to devise a notation providing less room for errors. One example is the notation for tables. The StarTeX code for generating table 1 is

```
<table>A small table sample
  <row> <b>Index</b> <col> <b>Data</b>
  <row> 12 <col> 199
  <row> 17 <col> 0
</table>
```

employing only one environment (<table>...</table>) and two simple commands: <row> starts another row, and <col> starts another column. As any combination of these two commands is legal, there will be fewer user errors.

2.5 Structural versus visual markup

Like LaTeX, SGML and HTML, StarTeX provides a system using *structural markup* in which the user indicates the structure of the document with (ideally) no regard to its appearance. This is in contrast to systems like FrameMaker and Word that emphasize the *visual* side of publishing.

While both kinds of systems have their merits and proponents, I believe that it will be an advantage for the students to have a working knowledge of structure-based text markup.

Since StarTeX was designed to cater for a very limited range of documents, it goes even further than LaTeX and provides absolutely no visually oriented commands. (Even though LaTeX is primarily a structural system, it does provide some visual commands like \vspace, \rlap, \raisebox and others which are necessary when fine-tuning the document.)

2.6 Processing speed

Even though LaTeX is not slow — it processes 4–10 pages per second on a Sun Sparc — it does take some time to start it. A sample 2½-page LaTeX document takes 2.8 seconds to run on a Sun SparcStation 20 and 8.7 seconds on a Silicon Graphics Indy. Starting times in this range are not important when writing long documents, but the potential users of StarTeX tend to have quite short ones. They also process their documents very frequently, as they are yet unfamiliar with LaTeX and want to see the effect of a command or test for an error.

Since StarTeX is much smaller than LaTeX and uses only one configuration file, it is faster to start. The sample document mentioned above takes 0.9 seconds on the SparcStation and 1.6 seconds on the Indy.

2.7 Error recovery

The philosophy of error recovery in TeX (and LaTeX) is based on user interaction: when an error is detected, TeX enters an interactive mode in which the user may attempt to locate the error and manually insert or delete code so that processing may continue. This technique was designed in the days when computers were slow compared to current ones: a 100 page document that takes 20 seconds to process today could take more than an hour 10 years ago.

So far, I have found no students taking advantage of this interactive error correction. They either terminate the run after the first error, or they let LaTeX run to completion hoping to find the error in the log file afterwards.

Running a batch-oriented text processor has a lot in common with compiling, so StarTeX tries to benefit from experience gained in this field.

Figure 2: A minimal StarTEX document

```
<body>
<title> <startex><--->A <tex> for beginners </title>
<author> Dag Langmyhr<p> Department of Informatics<p>
  University of Oslo<p> <tt>dag@ifi.uio.no</tt>
</author>
<info> <today> </info>

<abstract> This document describes <startex>, a special <tex>
  format for students writing their first project report.
</abstract>

<h1> The basic philosophy of <Startex> </h1>
<Startex> was designed for novice <tex> users. It employs a
different notation and a different set of commands from <latex>,
and the idea is that this makes it more user-friendly for these
users than plain <tex> or <latex>.

<p>
The notation used in <startex> resembles HTML and some of the
commands are the same, but the philosophy of the two is
different. HTML was designed to display hypertext information
on a computer screen, while <startex> is used to produce a
student report on paper.
</body>
```

It runs in non-stop mode, and attempts to recover from any errors it detects. Using named environment terminators makes this easier, so an error like

```
<h1>Using <b>bold text</h1>
```

is easily detected and rectified.

2.8 An example

Figure 2 shows an example of a minimal StarTEX document.

3 Comparison with SGML and HTML

Since the notation used in StarTEX is so similar to HTML, some users have wondered why I did not instead implement an HTML processor in TEX. There are several reasons for that:

- HTML is not yet stable. New versions are appearing regularly, and various companies introduce their own extensions. No-one knows what HTML will look like a few years from now.
- It has not been decided how advanced will be the support for mathematical typesetting in HTML.
- It is possible to define new user commands in StarTEX, and I think this is very useful when working with a system like StarTEX. There is no possibility for such definitions in HTML.
- It is difficult to write a robust parser for HTML in TEX.

It would, however, be possible to combine StarTEX and SGML (except for the problem with user-defined commands). One could envisage an environment in which the user edits his or her document using an SGML editor connected to a StarTEX DTD. Once the document is finished and verified by the editor, StarTEX could serve as its printing processor. Development of such a system is a future project.

4 Concluding remarks

StarTEX was created to help one particular group of students; it was *not* designed to replace LATEX for general work. Instead, it is a tribute to the versatility of TEX which so easily permits the design of a completely new user interface.

StarTEX was completed early this autumn. It is available by anonymous FTP from ftp.ifi.uio.no in the directory pub/tex/startex.

References

[1] Donald E. Knuth. *The TEXbook*. Addison-Wesley, 1984.

[2] Leslie Lamport. *LATEX User's Guide & Reference Manual*. Addison-Wesley, 1994. Second edition.

[3] Michael Spivak. *The joy of TEX*. American Mathematical Society, 1986. The guide to $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX.

[4] Philip Taylor. TEX: an unsuitable language for document markup? Talk given at the EuroTEX 1995 conference; does not appear in the proceedings., 1995.