

ドキュメントマークアップ言語：TeX

北川 和裕
斉藤 信男

慶応義塾大学理工学部数理科学科

TeXは、スタンフォード大学のKnuthが開発した高機能文書整形システムである。TeXは、文書の構造とその構成要素の関係を記述するシステムではなく、文書の要素に対して構成の指定を行う機械的な処理を記述したものである。そのため、初心者や専門家以外にとっては非常に使いにくいシステムとなっている。それを解決するためのTeXのマクロパッケージである、LaTeXをL.Lamportが開発した。LaTeXは、アレーンなTeXとは異なり、文書の構造を記述するものである。

本論文では、このようにTeXをドキュメントマークアップ言語の一つと考え、その特性を論ずる。またこれらをもとに、TeXのアロプログラミング方法論についても触れる。

Document Markup Language: TeX

Kazuhiro Kitagawa
Nobuo Saito

Department of Mathematics, Faculty of Science and Technology
Keio University

Abstract

TeX is a high quality document formatting system developed by Prof. Knuth at Stanford University. TeX consists of low level commands that describe the form of the document and all the logical structure components. It does not support the commands associated with a logical structure of document for the complete formatting of the corresponding concrete structure. So, it is awkward and difficult to use this, especially for the novice or casual users. To avoid this, L.Lamport developed LaTeX that allows the definition of the macros that specified the logical components of the document.

The paper presents the review of TeX from the point of the document markup language. Finally, the authors describe the programming method of TeX based on previous experience.

1.はじめに

TeX[6]は、高機能文書整形システムで、英文文書の出版の過程のほとんどを行うことができる。TeXは、プログラミング言語で言えば、アセンブラレベルのもので、そのコマンドは非常に低級なものである。そのため、初心者や専門家以外に取っては、使いにくいシステムとなっている。TeXの命令は、文書の構造とその構成要素の関係を記述するシステムではなく、文書の要素に対して構成の行う機械的な処理を記述したものである。即ち、TeXは、文書の論理構造を指定するものではなく、物理的な整形の指定を行うものである。マークアップ言語には、文書の物理的な構成を指定する手続き型言語と文書の論理構造を記述する宣言型言語の二つがあるが、一般にTeXは、前者に属する。TeXには、L.Lamportが開発したLaTeX[7]と呼ばれるものがある。LaTeXは、TeXのマクロとして実現されており、文書の構造を記述するものである。このようなことを考えるとLaTeXは、宣言型マークアップ言語として取り扱うことができる。

本論文では、手続き型マークアップ言語と宣言型マークアップ言語の特徴を考えながら、TeXをドキュメントマークアップ言語の一つと考え、TeX、LaTeXの特性を論ずる。最後にこれらをもとに、TeXのプログラミング方法論について触れる。

2.マークアップ言語と文書モデル

文書の整形の記述に用いられるマークアップ言語は、一般に手続き型マークアップ言語と宣言型マークアップ言語の二つに分けることができる。手続き型マークアップ言語は、実際の文書整形に必要な機械的な手続きを記述するものである。このような言語は、低級であるが非常に強力で、ユーザはこれらを用いて、文書の論理構造に基づいた制御命令を作成することができる。このような命令をマクロと呼ぶ。いくつかの手続き型マークアップ言語（SCRIPT[12]、TROFF[4]）では、マクロ命令を提供することによって、それらを宣言型のマークアップ言語として取り扱うこともできる。しかし、ユーザはこの段階でも低レベルな命令を使うことができる。このようなことを考えると、このようにマクロを提供しているマークアップ言語は、宣言型マークアップ言語として取り扱うことができない。手続き型マークアップ言語の欠点は、その言語使用が低級なため使いにくいことにある。その代わりに、ユーザは柔軟な整形の指定を行うことができる。

TeXは、低レベルな命令からできており、これらを用いて文書整形の指定を行うものである。従って、TeXは手続き型言語と言える。

宣言型マークアップ言語は、文書の論理構造のみを指定することによって、文書の整形を行うものである。全てのコマンドは、文書の論理構造に対応したもので、これらのコマンドが、自動的に整形をその指定に従って行う。宣言型マークアップ言語の欠点としては、言語の柔軟性にある。ユーザは、使いやすいが、細かな整形の指定を行うことが難しい。宣言型マークアップ言語には、Scribe[8][11]、GML[1]、SGML[2]などがある。

マークアップ言語としては、次のような点に留意すると宣言型マークアップ言語が使いやすい強力なものとなる。

- 1)柔軟なコマンド体系を採用し、ユーザの制限を緩和する。また、異なる文書構造の設定が容易に行える。
- 2)文書のレイアウトを記述したテンプレートを用いる。これらのテンプレートは、文書スタイル、ユーザの整形要求を容易に取り込めるように、整形に必要な変数（幅、ページ長、マージンなど）がパラメトリックに設定できる。

文書整形する場合には、文書の論理構造と物理構造の指定を行うことになる。この二つの関係を、図2-1に示す。図に示すように、論理構造と物理構造は基本的に独立に取り扱うことができるが、両者は密接な関係を持っている。即ち、文書の論理構造が決まれば、文書の物理構造は決まってしまう。

実際には、文書の論理構造とその階層構造から生じてくる構成要素の定義を行わなければならない。このためには、つぎのような定義が必要となる。

- 1)文書のサイズ
- 2)フォントファミリー
- 3)スタイル
- 4)文書の構造とそのレイアウト

この4)が、構造化文書と呼ばれるもので、文書の構成の指定を行う。1)、2)、3)、は、各要素の形式の指定を行うものである。これらが決って始めて、文書の整形を行うことができる。

整形システムとして、TeXのような手続き型マークアップ言語を選んだ場合には、これらは支援はされていないが、一意に決められてはいない。

このように、ユーザインターフェース、整形の過程を考えると、宣言型マークアップ言語を用いて文書を記述することが望ましい。

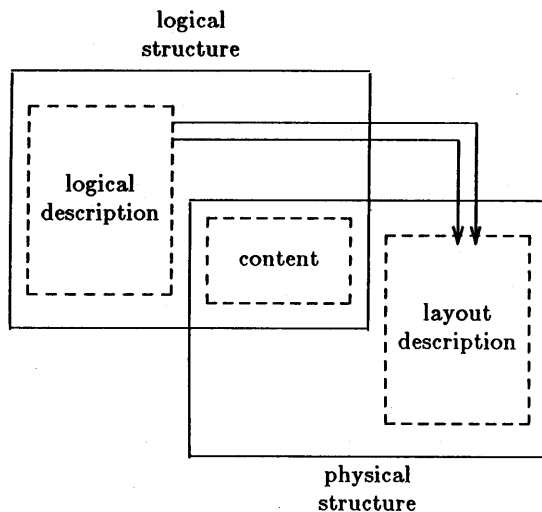


図2-1 文書の論理構造と物理構造

3. TeXとLaTeX

TeXは、それ自身では具体的な整形処理を行うことができない。実際のTeXの構成は、図3-1のようになっている。

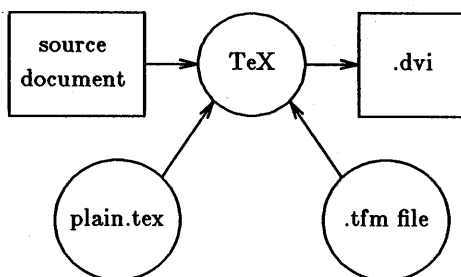


図3-1 TeXの構成

この図のplain.texには、TeXが整形を行うためのデータ、マクロなどが定義されている。また、tfmファイルには、フォントの情報が格納されている。TeXを実行する際に、常にplain.texとtfmファイルの参照を行うとその処理が非常に遅くなる。これを避けるため、TeXコマンドを実現する際にこのplain.texとよく使われるtfmファイルを前もってロードしておく。

一般に、plain.texがロードされたものをTeX (アレーンTeX) と呼ぶ。plain.texをロードする前のTeXには、100余りのコマンドしかない。これらTeXの持つ基本的な制御命令は、コマンドコードと呼ばれる。ま

た、整形を行うに必要なデータ、フォントの指定は行われていない。plain.texでは、この基本的なコマンドコードを用いてマクロを記述している。このようにplain.texには、実際の整形に用いられる初期値とその規則に関する情報、コマンドコードで記述したマクロが格納されている。このplain.texには、文書の物理的な整形を指定するものしか定義されていない。

TeXと同じ機能で、異なるコマンド体系を持つTeXを作りたいときは、plain.texを変えれば良い。実際には、LaTeXやAMSTeX[9]がこれに当たる。LaTeXは、plain.texの代わりにlpain.texをロードすることによって実現されている。TeXの場合にはplain.texを、LaTeXの場合にはlplain.texをロードして実現されているので、LaTeXでlplain.texで定義されていないTeXの命令を用いることはできない。また、その逆もできない。但し、両方に共通したTeX自身が持つコマンドコードはLaTeXでも用いることができる。

LaTeXは、TeXを使いやすくするためのマクロパッケージである。LaTeXがTeXと異なるところは、TeXが物理的な整形の指定を行っていたのに対して、LaTeXは、Scribeのように文書の論理構造を指定して文書の整形を行う。このように、LaTeXでは、文書形式を記述するTagを入力して整形を行う。また、文書の構成要素は、開始と終了を示すTagで囲まれる。

LaTeXの構成を、図3-2に示す。LaTeXには、多くのスタイルファイル(.sty)が用意されており、LaTeXのソースプログラムでそれらを指定すると、指定された文書スタイルに整形される。同じ名前のTagでも、文書スタイルが異なれば整形は異なる。また、LaTeXには、スタイルファイルとは別に、多くのスタイルパラメータが支援されている。これらを変えることによって、物理的な構成を変えることができる。即ち、文書の物理構成を変える場合は、スタイルファイルの指定とスタイルパラメータのみを変えるだけでよい。この場合、Tagを変える必要はない。

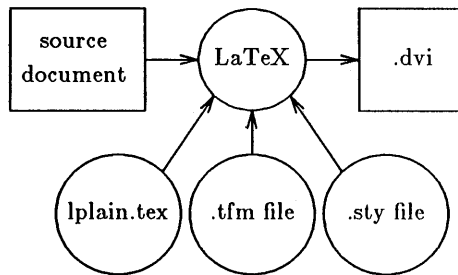


図3-2 LaTeXの構成

図3-3に、文書の論理構造を指定するTagの例を示す。また、図3-4にスタイルパラメータの例を示す。

```

\paragraph
\subparagraph
\chapter
\section
\subsection
\subsubsection
  
```

図3-3 Tagの例

またLaTeXでは、物理的な構造の指定となるフォントの指定、センタリング等は、Tagの環境を指定することで行うことができる。LaTeXやTeXで言う環境とは、文書のなかの特定の部分の属性を指定したものである。勿論、それらを行うためのコマンドも提供されている。TeXでの環境の設定は、{}で囲むことによって行われる。TeXとLaTeXの環境の設定の例を、図3-5に示す。この環境は、指定された範囲内でのみ有効である。

図3-6にTeXとLaTeXの実際のプログラムの例を示す。このように、TeXと言っても3つの階層がある。最も低次元なコマンドコード、コマンドコードを使い易く記述したTeX、文書の論理構造をもとに実現されたLaTeXの3つである。但し、LaTeXでも細かな操作を行おうとするとコマンドコードのレベルを取り扱わなければならない。

```

style parameter:
  \topmargin
  \headheight
  \headsep
font parameter:
  \topfraction
  \textfraction
paragraph and making parameter:
  \parskip
  \baselineskip
box:
  \fboxrule
  \fboxsep

```

図3-4 スタイルパラメータ

```

TeX:
  {\bf Bold Face}

LaTeX:
  \begin{bf}
    Bold Face
  \end{bf}

```

図3-5 TeXとLaTeXの環境の設定の例

```

TeX :
  \font\title = cmbx10 scaled \magstep3
  \leftline{{\title This is chapter.}}
  \vskip 1in

  \TeX system is ...

LaTeX :
  \documentstyle{article}
  \begin{document}
    .
  \chapter{This is chapter}

  \TeX system is ...

```

図3-6 TeXとLaTeXのアロプログラム例

4. TeXのアロプログラミング

2章で、宣言型マークアップ言語が使いやすい言語であるということを述べた。このことを考慮すると、通常の場合TeXよりもLaTeXを用いる方がよい。また、TeX自身が支援しているコマンドコードを用いることは、極力避け、LaTeXが提供するコマンドを用いた方がよい。例えば、センタリングはTeXのコマンドコードでは、

```
\hbox to\hsize{\hss{centering}\hss}
```

と記述する。これではユーザは、何をやっているか理解するのは困難である。

つぎに、実際のLaTeXのアプログラミングスタイルについて述べる。

1) 文書を分割し、階層を作る。

TeXには、外のファイルを取り込むコマンド”\input”がある。これを用いて、ファイルをタイトル、章毎に分割して格納する。また、各ファイル総てに共通に用いられるフォントの定義などは、全ての親であるタイトルを記述したファイル、または別のファイルに登録しておく。このファイルの例を、図4-1に示す。

```
\input keio           % macros and parameter for Keio Univ.
\documentstyle[12pt]{article}
\begin{document}
\hyphenation{dis-tri-but-ed}
\title{Document Markup Language:\TeX}
\author{Kazuhio Kitagawa} \\
\author{Nobuo Saito} \\
{\it Keio University} \\
\date{13 May 1987}
\maketitle
\vskip 1cm
\input intro         % introduction
\input chapter1     % chapter1 include section
\input chapter2
\input conclusion
\end{document}
\end
```

図4-1 タイトルファイルの例

2) 環境を用いる。

LaTeXでは、TeXで用いられる一般的な環境とLaTeXだけで用いられる環境があることは前に述べた。局所的に行われる物理的な整形の指定は、この環境設定を用いるのがよい。

LaTeXの環境とTeXの環境は、使い方は異なっているが全く同じものであるが、この二つを使い分けることが望ましい。単語、一行の一部に対する指定では、{}で属性の有効範囲を指定するTeXの環境を、環境の指定が複数行に及ぶ場合には、Tagで環境の有効範囲を指定するLaTeXの環境を用いると読み易いプログラムとなる。この環境の使い方の例を、図4-2に示す。この例では、イタリックフォントを指定している。

```
TeX:
{\it Here is a italic line \/}

LaTeX:
\begin{it}
  Here \\
  is a \\
  italic line.
\end{it}
```

図4-2 環境の使い分け

3) シンボリックな単位の指定を行い、物理的な単位を統一すること

物理的な移動量は、TeXではインチ、ポイントサイズ、センチ、ミリなどいろいろな単位を指定することが

できる。無用な混乱を避けるため一つの文章中で用いる単位は、一つに統一した方がよい。またLaTeXでは、文書スタイルに応じた値が、シンボリックに定義されているのでそれを用いることが望ましい。また、コマンドとこれらの値を組み合わせたものが、LaTeXでは支援されている。

```
\vspace{.1in}
```

```
\vspace{\smallskipamount} ≡ \smallskip
```

この例の\smallskipamountは、文書のスタイルが変わると自動的に変わる。これにより文書スタイルを変えた場合にも、これらの値は変える必要がない。

フォントに対応した物理的な指定は、ex、emを用いることが望ましい。exは、標準的なフォントの高さ、emは幅を表す。この例を、図4-3に示す。

```
\hspace{-2em}////////
```

```
\hspace{24pt}////////
```

図4-3 exとemの例

ex、emの値は、フォント毎に決められているので、フォント指定を変えてもその値を変えずに済む。

4) 複雑な処理は、boxとglueを用いる。

TeXは、boxとglueという概念で構成されている。一単語をboxとして取り扱い、単語間のglue、行間のglueを調節しながら処理を行う。

複雑な処理は、個々の要素毎に物理的な処理を記述することによって行うことができるが、このboxとglueの概念を用いて、各要素を一つのbox、このboxを大きな一つのboxとして取り扱い、このbox間の指定をglueで行うことが望ましい。後者の方が、読み易く、プログラムが行い易い。

以上のことに留意することによって、読み易くポータビリティの高いTeXプログラムを記述することができる。

5.終わりに

これまで述べたように、一般の文書を記述する場合にはTeXよりLaTeXを用いるのが望ましい。TeXを提供している所では、必ずLaTeXも支援しているのでLaTeXを使って不利になることはない。

この様にLaTeXは、非常に使いやすいシステムであるが欠点がないわけではない。文書のスタイルに対応した物理的な整形を記述したスタイルファイルの記述が、非常に分かりにくくできていることである。本来これらのスタイルファイルも、整形指定と、その属性とに分けて構築されることが望ましい。

参考文献

- [1] C.F.Goldfarb, "A Generalized Approach to Document Markup," SIGPLAN Notice, no. 16.6, 1981.
- [2] ISO, "Information Processing - Text and Office systems - Standard Generalized Markup Language(SGML)," Draft International Standard ISO/DIS 98879, 1985.
- [3] ISO, "Text Preparation and Interchange -Text and Office systems" DP 8613 ISO/TC97/SC18/WG3, 1985.

- [4] B.W.Kernighan, "A Typewriter-independent TROFF," Computer Science Tech. Report 97, Bell Laboratories, 1981.
- [5] G.D.Kimura and A.C.Shaw, "The Structure of Abstract Document Objects," SIGOA, no. 5.1-2, pp. 161-169, 1984.
- [6] D.E.Knuth, The TeXbook, Addison-wesley, 1984.
- [7] L.Lamport, LaTeX:A Document Preparation System, Addison-wesley, 1984.
- [8] B.Reid, "Scribe:A Document Specification Language and its Compiler," CMU-CS-81-100, CMU, 1980.
- [9] M.Spivak, The Joy of TeX, AMS, 1983.
- [10] TUGboat, TUG, P.O.Box 9506, Providence, RI 02940.
- [11] Unilogic Ltd., Scribe Document Production Software User Manual, 1985.
- [12] Waterloo, SCRIPT Reference Manual, Dept. of CS., Univ. of Waterloo, Ontario.