論文形式の LaTeX 文書のスライド発表形式の LaTeX 文書への変換プログラム

日本 IBM(株)　東京基礎研究所，金沢工業大学　客員教授　寒川　光

2007 年 9 月 20 日

『LaTeX & PostScript スーパーユーザのテクニック』で，論文形式の LaTeX 文書を，Perl による簡単な変換プログラムによって，スライド発表形式の LaTeX 文書に変換する方法を紹介した[1]．スライド発表用には LaTeX 2ε の seminar クラスを使用し，変換には Perl プログラムに対する指示文（ディレクティブ）を使用する．本資料は「数学ソフトウェアとフリードキュメント V」での講演資料として，このアプローチを発想した周辺事情を含めて紹介し，講義や勉強会などで数式を多用する方々にこの方法を利用していただく目的で用意した．

指示文の利用 Fortran や C 言語によるプログラムでは，原始（ソース）プログラムにコンパイラに対する指示文（言語規格上ではコメント行だがコンパイル時には特別な指定をできるステートメント）によって，2 通り，3 通りにコンパイルする方法が利用されている．この方法と同様に，LaTeX のソーステキストを原始プログラムと見立てて，シングルソースで論文用と発表用を 2 通りに変換されるようにした．この場合，変換後のテキストには直接手を加えない方針とする．両者に共通のソースを用いて，改訂はそれに対してだけ行えば両方に反映されるようにする．

スクリプト言語プログラミング 数値計算には Fortran，システムプログラミングには C 言語，Web サービスでは Java+XML というように，処理内容に適したプログラミング言語がある．文書の変換には Perl の正規表現を利用すると，意外に簡単に変換プログラムを実現できる．

仕様の決定と改良 実際に使用しながら，自分のプレゼンテーションに適した形に仕様と変換プログラムを改良してゆく（**日曜大工プログラミング**の楽しみ）．

1　seminar クラスの使い方

クラス名を seminar としてスライドにしたいものを “\begin{slide}” と “\end{slide}” で囲う．図 1 は，ドキュメントクラスに “seminar” を指定して，「\LaTeX の picture 環境での作図機能は次のように纏められる.」の 1 センテンスの後に enumerate 環境によって 4 項目を箇条書きした例である．ソーステキストのファイルを semitest.tex とすれば，“platex semitest” と “dvipsk -t a3 semitest” で semitest.ps を作成できる．オプションの “-t a3” は dvipsk に与える用紙サイズである．“\slideframe{none}” を指定すれば外側の枠を消すことができる．

2　論文用テキストからスライド用テキストへの変換

ソーステキストが **1 センテンス 1 レコード原則**で書かれていると，変換プログラムは例えば Perl なら “while(<F>){...}” のループ構造の中で処理しやすい．入力ファイルの 1 レコード単位に文脈と指示行に

[1] 共立出版，2006 年 11 月．“http://www.kyoritsu-pub.co.jp/shinkan/shin0611_07.html”

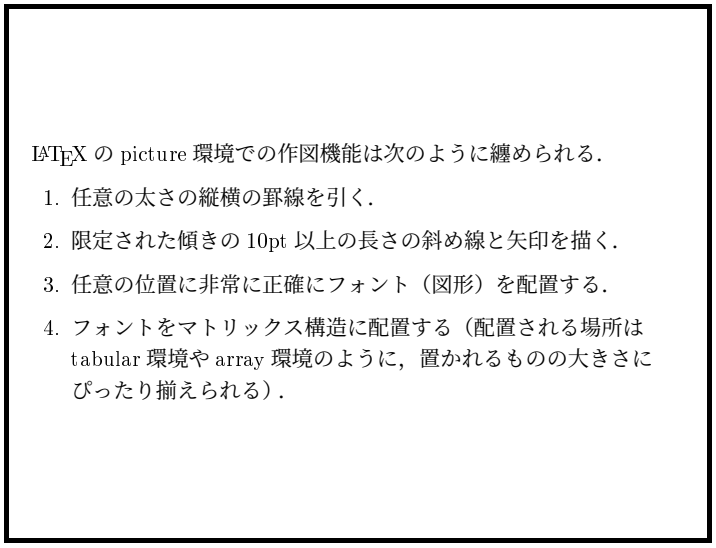

図 1: seminar クラスによるスライドの例

従ってレコードを加工すればよいからである．

プログラム名は texslide.pl とした．hoge.tex ファイルを変換して，slhoge.tex を作成する場合には “perl texslide.pl hoge.tex > slhoge.tex” とする．このあと “platex slhoge” と “dvipsk -t a3 slhoge” とするので，makefile を用意して “make” で論文用が，“make slide” でスライド用ができるようにしている．

移行対象の環境　figure 環境，equation 環境，table 環境などを検出し，この環境をまとめて “\begin{slide}” と “\end{slide}” の中に埋め込む．変換対象の環境は figure，table，equation，displaymath，eqnarray，align，gather，quote，itemize，enumerate，description，thebibliography 環境としている．検出は入力行から \begin{xxx} を探しており，1 文字目にブランクを入れた場合は変換対象から除外される．

ダミー環境　論文には書かれていないが，プレゼンテーションでは必要な（式の細かな変換など）補足説明のためのスライドや，論文の紙数制限で割愛した数式を表示する目的で使用する．また大学の講義では，先週の講義で使用した図をその日の講義テーマの前に復習で見せるなど，別ファイルの図を説明に使いたいということがある．これらの目的で，上記の環境の行頭に%を 1 つだけ付加した行は，%を外してスライドには現われる機能を設けた．

見出しの表示　section，subsection，subsubsection，paragraph に対しては，その引数（見出し）をそれぞれ Huge，huge，LARGE，Large のフォントサイズでスライドの中央に center 環境で出力するように変換している[2]．

ディレクティブ　上記の変換だけでは不十分であるが，次の 4 種類のディレクティブを挿入することで，実用的なスライドの加工が可能になる．

%term　キーセンテンスの直前に “%term” の 1 行を加える．これによって次のレコード（行）を itemize 環境のアイテムとして埋め込む．1 センテンスではなく，センテンスの一部や**単語**を指定する場合は，1 センテンス 1 レコード原則を諦めて，改行を細かく入れて，表示に回す部分を 1 レコードと

[2] スタイルファイルとして jarticle.cls を対象としている．

して切り出し，その直前に%term を指定する．指定されたキーセンテンスは見出しの下に箇条書きされる．

%newslide 指定したキーセンテンスが多いと，1 枚のスライドに入りきらない．そこで “%newslide” を入れて強制的に**スライドを改める**ようにした．

%copystart と%copyend このディレクティブに挟まれたソーステキストは，スライド用のソーステキストに**コピー**する．複数の数式を 1 枚のスライドに収めるために使用する．講義などで，式の展開を確認しながら説明する場合には必須である．

% SlideReplace seminar 環境での図のサイズ調整のために用意した．図を縮小する必要がある場合，これを変換プログラムで自動的に行うのは無理と考えた[3]．ここでは図ごとにサイズを，論文用とスライド用の 2 通りに指定して，“\psset{unit=1cm} % SlideReplace\psset{unit=8mm}” のように**置き換え**る．

3 『LaTeX & PostScript スーパーユーザのテクニック』例

共立出版さんのご厚意によりサンプルをダウンロード可能としていただいた．これらは金沢工業大学でのプログラミング演習の説明や，雑談などの目的で用意したものが大半であるが，数式が含まれるものが多く，LaTeX 以外のツールで説明文章を用意することに気が進まないものである．説明文章とプレゼンテーション用スライドを 2 重に用意するのは手間と考え，これらの資料の作成と並行して，texslide.pl を改良していった．変換プログラムに print ”\\slideframe{none}\n”; の 1 行を加えることで，%size ディレクティブを不要にできる．

ベジェ曲線 コンピュータグラフィックスの内容で，フォントデザインには欠かせない知識である．内容に数式の展開が多いので，スライドは %copystart と %copyend を多用して，1 枚のスライドで式の展開が追える形にした．

RSA 暗号の根拠となる表 $x^k \pmod m$ を，m や k を変更して作表する TeX マクロ（アライメントタブを生成するマクロ）を解説した．後半は $x^k \pmod m$ に関する話題（フェルマーの定理）と，RSA 暗号，またユークリッドの互除法と連分数の幾何学的解釈について述べる．

Calcomp インターフェース 古典的なペンプロッタのインターフェースを，PostScript（calcomps.c）と X-Window（calcompx.c）で実装する．どちらも Fortran と C から呼び出し可能である．

対話型プログラム Calcomp インターフェースでは計算結果のポスト処理をしたので，X-Window の対話型の機能を利用しなかった．ここではその延長で対話形式の機能（イベント駆動型のループ構成）を経験してみる．

フラクタル Cygwin や Linux 環境でのプログラミングと計算結果の出図の方法を実習する．題材はフラクタル図形で，その語源である分数（fractional）次元の説明も加えた．

疎行列のプロット 数値解析プログラムが差分法と有限要素法を使用した場合の，疎行列の非ゼロ要素の位置と，その行列を三角分解したときに現れるフィルインと呼ばれる要素の位置を作画する．

MPI ライブラリ IBM の超並列型スーパーコンピュータ Blue Gene で，MPI ライブラリを MPICH2 をベースに実装した方法を解説する．ここでは説明文に記述された以上の図をスライドに用意して，MPICH2 に少しずつコンポーネントを追加してゆく様子を紙芝居でプレゼンテーションする．

[3] 変換対象の環境の終わりまでを読み込んで，スライドに変換された場合の縦横のサイズを予測する必要がある．

2 次元配列分割・分散 線形代数の並列計算では行列を 2 次元配列分割して，並列計算のノードに分散させる．密行列を 2 次元ブロックサイクリックに配列分割して MPI プログラミングする人には便利なデバッグツールでもある．

写真機のレンズ レンズ面を増やすと収差を除去するためのパラメータが増えるので，収差の少ないレンズを設計することができる．初等関数の多項式近似で，多項式の次数（パラメータ）を増やすと精度が良くなるのに似ている．Calcomp インターフェースの演習問題の背景の説明でもある．

住所録から宛名変換 ソーステキストの変換プログラムの概要を述べる．宛名の郵便番号は正確に置く必要があるので，TEX の picture 環境で座標値を指定している．

MusiXTEX 楽譜を書くための MusiXTEX を紹介し解説する．「楽譜とその演奏」を「文書テキストとその印刷」のアナロジで考えたエッセイを付加した．「音律は音楽のアーキテクチャ」か．

4 おわりに

情報は上流から下流へとコンパイルされ変形されるたびに形を変え，最終的には美しい図を含む読みやすい文書になるが，ある種の情報，例えば図形の持つ論理的な意味をプログラムで抽出することは難しくなる．ここでの「意味」は，対称性や繰返し，曲線の次数などを指している．このような意味は，資料の最上流では可読な形で存在するが，下流へ進むたびにそれはプログラムの実行結果に置き換えられてしまうので，上流のほうが意味を正確に扱うことができる．最終的には PDF ファイルが表示したビットマップを人間が眺めて判定するしか方法がなくなる．この段階まで進んでしまうと，GUI を使ってカットアンドペーストでパワーポイントファイルに貼りつけたりしなくてはならなくなる[4]．論文などの文書に描かれた図をスライド用にするには，拡大して配置し直すだけでよいので，ソーステキストのまま取り出せれば簡単にできそうである．数式や表も同様である．LATEX の場合，figure，table，equation 環境で書かれた図，表，数式を，拡大してスライドの中央に置くだけでよい．

『LATEX & PostScript ...』は金沢工業大学の大学院で「特別講義」として行ったテキストを元にしている．2002 年に始めた講義内容で，当時はグリッドコンピューティングが脚光をあびていた時期で IT 革命の進行とともにシステムが複雑化・巨大化してきた時期である．この複雑化を支えてきたシステムの階層構造と**マルチプログラミングリンガル**の重要性を痛感していた．

論文や書籍出版などを扱う「文書処理」アプリケーションでも，ハードウェア（写植機），制御言語，その上に構築される LATEX 2ε と PostScript の世界があり，文書はその上に書かれてゆく．これらの階層を縦に繋げて，どの階層で何がおこっているかを仔細に分析する経験は，システムエンジニアとしてエンジニアリングセンスを磨く上で役立つのではないかと考えた．ここには普段はあまり目にしないマクロプログラミングや PostScript 言語に触れる新鮮さもある．また，プログラムの仕様を決める経験もできる．この題材では，自然にプログラミングを原理的な姿から考え直す機会を与える．

インターネットの上に展開された IT システムの階層構造はとても深くなり，各階層に現在でも次々と新しい技術が投入されている．アセンブリ言語で機械語を並べるプログラミングを平屋にたとえるなら，現代の IT システムは高層ビルディングにたとえられよう．その最上階からアプリケーションを叩く（に要求を出す）と，下の階でアルゴリズムが回って複雑な処理がなされる．システム全体をブラックボックスとして見るのではなく「なぜこれが機能するのか」という問題意識を忘れずに，アルゴリズムは何階で回っているのだろうかと考える習慣を忘れないでほしい．1 つ下の階のことは考えれば分るが，2 つ下の階になると，十分な知識がなくては分らないことが多い（各階のアルゴリズムは意外に単純な場合が多いにもかかわらず）．

[4] LATEX のソーステキストで eps ファイルで取り込まれた図を，LATEX でコンパイルして PS ファイルに変換し，さらに PDF 変換してからそれを巨大なビットマップファイルとして取り出してパワーポイントに貼りつけたりすると，**クリック猿**と呼ばれかねないので注意しよう．