

Design of Presentations: Notes on Principles and L^AT_EX Implementation

Boris Veytsman

Email borisv@lk.net

Address Computational Materials Science Center, MS 5A2
George Mason University
Fairfax, VA 22030

Abstract There are many T_EX packages available for creating presentations. Mostly they imitate the ubiquitous style of a certain tool, striving to produce Power-Point-like slides, hopefully with better typographical execution.

In this talk the principles of good design for presentations are considered. We discuss the problems with the common design of presentations as well as the famous proposition by Tufte to avoid slides at all. We try to formulate the principles of good presentation design and discuss T_EX implementations from this point of view.

The discussion is based on the author's experience in making slides for talks, lectures and training sessions.

1 Introduction

Probably one of the most often asked questions in news groups like `comp.text.tex` and other T_EX forums is this: "I want to create a nice presentation like my colleagues do, but I want to have beautiful math. Is it possible to make one using L^AT_EX?". The indispensable T_EX FAQ [1] recommends a number of possible answers to this question at <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=slidecls>. However, this is in my opinion the wrong question. It is akin to the question, "How can I create texts like my colleagues using word processors do?" Of course you can do this (and CTAN has a package for such task!), but why would you want to add an ugly text to so many ugly texts already existing?

The right question, in my opinion, should be structured differently. One should ask first, "What kind of presentations should I make, if any?" Only after this question is answered, one may think about the tools to create the right kind of presentation.

Here, as in many other fields of Computer Science, the approach should be goal-oriented rather than tool-oriented.

This paper tries to answer the first question and describe the “right kind of presentation”. Then we outline the tools to create such presentations in the L^AT_EX document preparation paradigm.

It should be noted that this paper describes the personal experience and opinions of the author. They are necessarily subjective and reflect the author’s idiosyncrasies and tastes. Therefore I recommend that the reader take my conclusions with the usual grain of salt. This paper is intended to be a starting point for the reader to think about presentations rather than an exhaustive and balanced guide to presentation design and implementation.

2 The Bane of Slides

If we judge by the folklore of office dwellers, presentation slides are an instrument of torture comparable to the tools of the Inquisition. A Google search for “Death by PowerPoint” gives about 5.6 million links [2]. The leading expert in graphical design, Edward Tufte, makes a convincing case to banish slides altogether [3]. He shows that the misuse of viewgraphs leads to information obscuring and shows how this hiding is at least partially to blame for the two tragic NASA catastrophes during Challenger and Columbia flights. The author of this paper has had his share of listening to presentations accompanied by slides, and can attest to the detrimental effect of the projector on the audience’s attention and the level of discussion. Unfortunately the “culture of PowerPoint” exposed by Tufte is spreading. One of my students, a federal contractor, complained that after his company submitted the results of the engineering study ordered by an esteemed US agency, the customer asked him to supply “PowerPoint slides with the results of the study”. “Our white papers had all the data and conclusions in a form suitable for engineers,” lamented the student, “but they wanted a dumbed-down version on a dozen slides”.

Tufte argues that the cognitive style of Power Point presentations has the following characteristics: “foreshortening of evidence and thought, low spatial resolution, a deeply hierarchical single-path structure as the model for organizing every type of content, breaking up narrative and data into slides and minimal fragments, rapid temporal sequencing of thin information rather than focused spatial analysis, conspicuous decoration and Phluff, a preoccupation with format not content, [and] an

attitude of commercialism that turns everything into a sales pitch” [3, p. 4]. This leads to his advice to abolish slides and use printed handouts instead. He quotes a funny parody of the Gettysburg Address where the moving speech was turned to a series of boring items and meaningless charts [4].

This is a drastic recommendation. It is tempting to declare slides completely broken and return to the good old days of great speeches. However we should remember that in those days Gettysburg Addresses were not the sole oral genre. There were university lectures, engineering reports, and other oral presentations that required some visual materials. We discuss this and the lessons learned in the next section.

3 The Lessons of the Blackboard

Almost any lecture in mathematical or physical sciences since ancient times was accompanied by chalk and a blackboard. Equations were derived on the blackboard. Important theorems were written down and often a frame was drawn around the most important ones. The professors teaching humanities used blackboards less often, but some of them wrote important points and conclusions on the blackboard. It was considered to be an important teacher’s skill to have a clear and logical blackboard at any time during the lecture. The author of this note, by the way, had a chance to study the art of teaching in the pre-slide days, and was often scolded by the professors because his own blackboard notes were not designed well enough. Obviously the importance of the visual accompaniment to a lecture was self-evident to the professors.

What lessons can we learn from classical blackboards?

First, the blackboard was always an auxiliary, a tool, but never the centerpiece of the lecture. It was unthinkable for a lecturer to start with the blackboard design, and then create a lecture around it. Even stranger would be the idea of writing the lecture on the blackboard, and then reading it from there. A lecturer worked in a well-lit room, and the attention of the audience was on him, not on his board. This is well captured by the famous photograph by Tarasevich [5]. What a contrast to a modern-day presenter, working in a dark room, the screen with projected slides being the only bright spot, and the silhouette of the presenter himself almost lost in the shadows!

Second, the bane of many slides, the bulleted list, did not work well on the blackboard, and was used sparingly if ever. Actually, an itemized list is not a good way

to convey information in general, not just in presentations. One of the unfortunate decisions of Leslie Lamport in the early L^AT_EX design was probably to provide an easy way to create itemized lists [6]. Too many authors overuse and misuse such lists.

Third, there is a very important issue of the speed and rhythm of a presentation. The text written on a blackboard appears slow enough for the audience to understand it. This is especially true for equations: the process of derivation of formulas in real time has an enormous education potential. In contrast to this, the information on slides often appears too fast, and the audience often cannot grasp it, especially complex equations.

These simple considerations lead to the following principles of presentation design.

4 Principles of Presentation Design

The first principle of presentation design is: if you can do without slides, do not use them during the presentation. The blackboard provides a good rhythm and tempo to lecture from. This rule is almost always applicable to lectures or training sessions, where a deep understanding of material is required. The situation is different for review presentations, where the aim is to give the audience a general picture of the subject and to stimulate an interest in the subject. In this case the breadth trumps the depth, and slides provide a better vehicle than a blackboard.

Slides may serve another useful function: they can be given to the audience to keep after the presentation. A better alternative would be to give the audience preprints instead [3], but this is not always practical. This conference (PracT_EX 2006) provides an interesting case for comparing these two forms of materials. Authors were urged to provide preprints to be included in the program, which will eventually be published in TUGBoat. These preprints take much more time and effort to prepare than slides with their “choppy” style. One could argue that they serve the audience better. Nevertheless, slides in some cases can be a cheap and efficient substitute of printable papers and manuals, as long as they are used sparingly.

This leads to the second principle: good slides must be designed for on-screen viewing by the audience after the presentation. This means that slides may (and often should) have such typographic attributes as epigraphs, footnotes and references; if they are not necessary for the presentation itself, they will be of use to the interested audience after the lecture.

The third principle is that the slides, even if used, are not the centerpiece of the

presentation. The most important part of the lecture is the body of thoughts the lecturer is trying to convey; the structure of the slides is secondary to these thoughts.

While this principle seems to be self-evident or even trivial, it is easy to point out many situations where it is violated. One of the transgressions is to divide the presentation into self-contained chunks exactly one slide long. Some slide making software even writes the name of slide on the top (or uses ugly “Continued...”), thus urging authors to think in terms of slides. Ideas are naturally mapped into a traditional hierarchy of sections (and sometimes subsections, etc.), and the length of the section should be determined by its content, not by the length of the slide.

If we follow the lessons of the blackboard, we should use step-by-step building of slides, comparable to step-by-step writing on the blackboard. Unfortunately this is overused by many authors, who, as noted by Tufte, reveal a line of their slide, read it aloud, than reveal another line, etc., making a bad design even worse. A more fortunate example is shown in Figure 1. The last diagram there looks intimidating for a non-TEXnical audience, and might shock it into inattention. A step-by-step revealing of points with a discussion of each is a better way to explain the usual TEX work flow, especially if accompanied by a judicious use of color.

Another transgression is the overuse of itemized lists, which is already mentioned above, and which is imposed on the authors by some slide making software systems. The structure of an itemized list is seldom good for a thought of any complexity. The usual misuse of dingbats as bullets makes these lists especially ugly.

A very rare example of a justified use of itemized lists and dingbats is shown on Figure 2. Note that in this case the bullets actually convey some information to the reader¹.

5 TEXnical Notes

This part of the paper is based mostly on the experience of the author, and even more than the other parts reflects his own tastes and preferences. I try to explain what turned out to work for me, and do not claim to make a comprehensive list of tools and methods available (see TEX FAQ for examples and comparisons) or state that my choices are necessarily the best ones. This is why I illustrate these ideas with

1. The bullets are based on the images distributed with the Debian package `gnome-extra-icons` under GPL

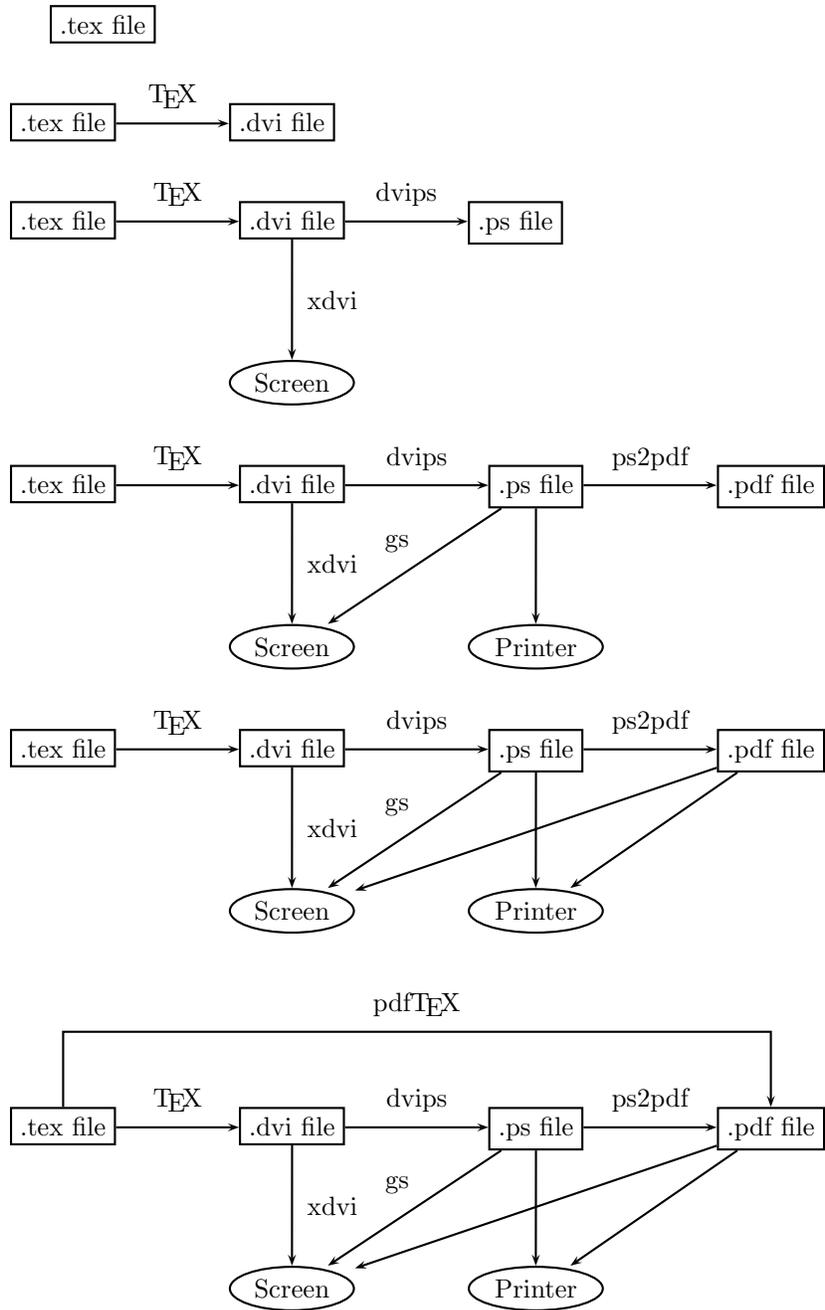


Figure 1: Step-by-step Building of a Flowchart

There are several kinds of users in Unix:



“Normal users”. These are people. John Doe \mapsto jdoe. The system knows a “real name” and *login*.



“System users”. They are owners of services: web service, time service, ftp service, etc.

Services are run by special processes, usually started up at boot up. These processes are called *daemons*.



Superuser or root. The classical system has exactly one superuser, who is allowed to do everything. It is used for system maintenance only.

Figure 2: A Rare Example of Acceptable Itemization

examples of my own presentation. I do not claim these presentations are especially good—they just reflect my approach to the design.

The remainder of this section is organized in chronological order: I describe the history of my journey into T_EX-based presentations and explain how I chose my tools and which lessons I learned.

The traditional `slides` class of L^AT_EX [7] was never sufficient for me because it has the `slide` environment, and I never liked to think in terms of slides. For the same reason I was not successful with the `Beamer` class [8] either. While this class allows sectioning, the `frame` environment and explicit overlays are a little too low level for my taste. I wanted the separation of the text into slides to be automatically generated by the sectioning commands, with some manual adjustments using `\clearpage`. This requirement was motivated mainly by laziness, of the kind described by Larry Wall [9]: I wanted the transition from slides to preprints and articles and back to be as smooth as possible. FoilT_EX [10], while lacking the sectioning, allowed the text to “overflow” slides, and this system was my initial choice. The addition of sectioning commands to this package was relatively simple [11]. I used this scheme for half a decade. When I started (it was the previous millennium!), PDF format was not as widespread as now, so I used FoilT_EX to format the transparencies used in the classroom, and L^AT_EX2HTML [12] to create the Web pages for the students to use after class. This is how the course notes [13, 14] were made.

Two Ways to Use Shell

There are two ways to write error-free programs; only the third one works.

Interactive: you write something, and the system executes it immediately.

You need:

1. Good process control mechanism
2. Good history and command line editing facilities

Batch: you write a script, and the system executes it. You need:

1. Good programming facilities (language!)
2. Good redirection facilities



2

Figure 3: A PDF Slide Created With FoilT_EX [15]

HTML provides very limited facilities for typography. PDF format is definitely better suited for online presentations and slides. One of my first experiments with slides in PDF format, suitable for class and online viewing, is [15]. It was done in FoilT_EX with pdfT_EX. A slide from this presentation is shown on Figure 3. The slides made in this way are quite readable and easy to prepare. However, there are several problems with FoilT_EX. First, it is not easy to prepare step-by-step slides. Second, FoilT_EX imposes too much white space on slides, which is not well suitable for online viewing. The density of the material is too low. Another concern is navigation tools. They are present in the slides [15], and FoilT_EX allows them to be placed easily on the bottom of the page, but not to the left or to the right.

A good question to ask is whether we need navigation tools as well as logos on the slides? They take up space without adding much to the information content. Tufte [3] puts logos on slides into the “chartjunk” category. Most viewing software has

a navigation menu, so a navigation panel seems to be a useless duplicate. However, a case could be made for navigation panels on slides. First, many people prefer to view slides in full screen mode, and adding a panel helps in this case. Second, the viewing software provides a generic navigation bar; the presentation author may want to use the customized one. Third, the proportions of a usual slide are wrong: they are wider than tall, and the text width is too large by any measure. By the way, Foil \TeX logos make the situation even worse: by design they subtract from the height of the page. The usual Web pages have the same problem of too large text width. Good Web designers create wide right or left margins on their web pages. A navigation panel with the logos to the left or to the right of the slide would serve the same function as wide margins in books, journal papers or good Web pages.

Last but not least, it is good to give the viewer a general idea where he or she is in the presentation or lecture. Traditionally, lecturers wrote the current topic (section or subsection) on the blackboard and changed it when they moved from topic to topic. Common presentation software tries to achieve the same purpose by adding a title to each slide. This is a bad idea because it imposes the rigid scheme “one thought, one slide”. The ugly scheme, “Topic blah on the first slide and Topic blah (cont.) on the subsequent ones,” only stresses the inadequacy of this design. Beamer [8] provides another solution: it allows adding copies of the table of contents before each section with the current topic highlighted. While this is definitely better than titles on slides, there are still disadvantages to this approach: the flow of the lecture (or on-screen viewing) is interrupted by basically the same page. Also, this table of contents is visible only at the section breaks, not in the middle of a section. A discreet reminder in the form of short table of contents on the wide margins of a slide seems to be the ideal solution.

An astute reader could note that the last paragraph describes the approach of pdfscreen package [16]. This package provides powerful facilities for on-screen presentations. After I discovered and tried this package, it was clear that I had found what I was looking for. I have been using it for the last several years. Some examples of its use can be found in [17, 18] (see Figures 4 and 5).

It is worth noting that all lectures except the first in the course [18] were taught using the blackboard only: the slides were made for the students to view after the lectures (when working on their homeworks and exams). The first introductory lecture was the only exception: it was taught using a projector. The aim of this lecture was to deliver a “general picture” rather than teach detailed knowledge.

George Mason University

School of Computational Sciences

Thermodynamics . . .

Closed Loop . . .

Computer . . .

Conclusions

Home Page

◀ ▶

◀ ▶

Page 6 of 19

Go Back

Find

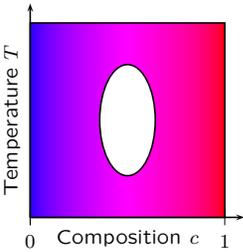
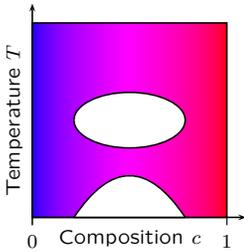
Full Screen

Close

Quit

2. Closed Loop Phase Diagrams

Guaiacol-glycerol; β -picoline-water, many polymers¹:

Van der Waals forces cannot produce this—there must be something else

¹T. Narayanan and A. Kumar, "Reentrant Phase Transition in Multicomponent Liquid Mixtures," *Phys. Reports* 249 (1994): 136–218.

Figure 4: A PDF Slide Created With pdfscreen [17]



School of Computational Sciences

- One Barrier
- Complex Transport ...
- Two Barriers And ...
- Homework

NANO500

◀ ▶

◀ ▶

Page 2 of 16

Go Back

Find

Full Screen

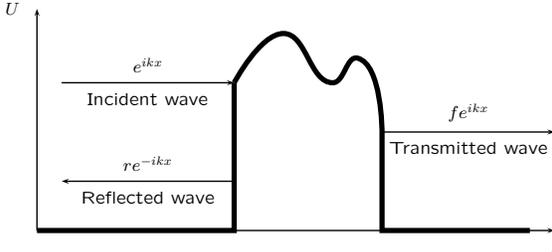
Close

Quit

1. One Barrier

1.1. Transmission And Reflection

Classical mechanics: particles are reflected by a barrier *or* penetrate it. Quantum mechanics: they do both.



Incident current $\propto 1$. Transmitted current $\propto |f|^2$. Reflected current $\propto |r|^2$.

Flux Transmission: $K_f = |f|^2$. Reflection Coefficient: $K_r = |r|^2$. Current conservation:

$$K_f + K_r = |f|^2 + |r|^2 = 1$$

Our goal is to calculate flux transmission and reflection coefficient.

Figure 5: A PDF Slide Created With pdfscreen [18]

```

%.pdf: %.tex %-pics.pdf
$(RM) $*.toc
pdflatex $*
- bibtex $*
pdflatex $*
while (grep -q \
'Label(s) may have changed'\
$*.log) do pdflatex $*; done

%-pics.pdf: %-pics.ps
ps2pdf $<

%-pics.ps: %.dvi
dvips -Ppdf -o $@ $<

%.dvi: %.tex
latex $<

```

Figure 6: A Fragment of a Makefile for Use of PSTricks and PDF Slides

There are several noteworthy details in the usage of `pdfscreen`. First, the default fonts for this package are serifed ones. I think it is better to use sans serif fonts for presentations, since they look better at low resolutions. The package `tpslifonts` [19] provides well designed and readable fonts, which are my personal favorites. Another package by Lehmke, `TEXpower` [20] is useful for providing step-by-step slides and overlays (although its page transitions are too glitzy for my taste).

Slides often require graphics. There are many different programs to generate them. Again, my personal favorite is `PSTricks` [21]. A package like `ps4pdf` [22] is useful for adding `PSTricks` pictures in slides. A fragment of a makefile that automatically generates all the intermediate steps is shown in Figure 6.

6 Conclusions

My experience of teaching and making presentations shows that slides are usually a bad idea both for in-class use and online viewing after class. In the first case slides cannot compete with the blackboard, and in the second case a good writeup is better than slides. Still, in some situations slides still can be used: for example, a need to show breadth rather than depth, a subject requiring many pictures, etc. It is important to remember in this case that slides must be adapted to both in-class and after-class use, and that slides are a tool, not the centerpiece of the lecture.

L^AT_EX tools provide ample opportunity to produce beautiful and instructive slides. My attempts to do this, documented above, show the power of these tools.

References

- [1] UK T_EX Users Group. UK list of T_EX frequently asked questions. <http://www.tex.ac.uk/cgi-bin/texfaq2html>, 2006.
- [2] Star Muir. Powerpoint use and misuse: Strategies for effective teaching. Lecture at George Mason University, March 2006.
- [3] Edward R. Tufte. The Cognitive Style of Power Point. Graphics Press LLC, Cheshire, CT, 2003.
- [4] Peter Norvig. The Gettysburg Powerpoint presentation. <http://www.norvig.com/Gettysburg>, January 2000.
- [5] Vsevolod Tarasevich. Duel. Sovetskoe Foto (Soviet Photography), (11), 1977. Available at <http://sov-photo.livejournal.com/35278.html>.
- [6] Leslie Lamport. L^AT_EX: a Document Preparation System. Addison-Wesley Publishing Company, Reading, Ma., 2 edition, 1994. Illustrations by Duane Bibby.
- [7] Frank Mittelbach. Producing Slides with L^AT_EX₂ ϵ . August, 1997. <http://ctan.tug.org/tex-archive/macros/latex/base>.
- [8] Till Tantau. User's Guide to the Beamer Class, Version 3.06, October 2005. <http://ctan.tug.org/tex-archive/macros/latex/contrib/beamer>.

- [9] Larry Wall, Tom Christiansen, and Randall L. Schwartz. Programming Perl. O'Reilly & Associates, Inc., Bonn; Cambridge; Paris; Sebastopol; Tokyo, second edition, 1996.
- [10] Jim Hafner. The Foil \TeX Class Package, October 2002. <http://ctan.tug.org/tex-archive/nonfree/macros/latex/contrib/foiltex>.
- [11] Boris Veytsman. Sectioning Commands in Foil \TeX and Conversion to HTML Format: Foilhtml Package, February 1998. <http://ctan.tug.org/tex-archive/macros/latex/contrib/foilhtml>.
- [12] Nikos Drakos and Ross Moore. The \LaTeX 2HTML Translator, March 2005. <http://www.ctan.org/tex-archive/support/latex2html>.
- [13] Sanat Kumar and Boris Veytsman. Mathematical methods in materials science (matsc597b). <http://www.plmsc.psu.edu/~www/matsc597>, 1996.
- [14] Boris Veytsman and Michael Kotelyanskii. Statistical thermodynamics of materials (matsc597c). <http://www.plmsc.psu.edu/~www/matsc597c-1997>, 1997.
- [15] Boris Veytsman. Introduction to unix. <http://users.lk.net/~borisv/unix>, 2001.
- [16] C. V. Radhakrishnan. Pdfscreen Manual, July 2000. <http://ctan.tug.org/tex-archive/macros/latex/contrib/pdfscreen>.
- [17] Boris Veytsman. Closed loop phase diagrams in liquid mixtures: From theory to simulations. <http://users.lk.net/~borisv/20040902lecture.pdf>, September 2004.
- [18] Boris Veytsman. NANO 500: Introduction to nanomaterials and interactions. <http://mason.gmu.edu/~bveytsma/nano500>, 2006.
- [19] Stephan Lehmke. Package Tpslifonts: Configure Presentation Fonts, July 2004. <http://ctan.tug.org/tex-archive/macros/latex/contrib/texpower/tpslifonts>.
- [20] Stephan Lehmke and Hans Fr. Nordhaug. The \TeX Power Bundle, April 2005. <http://ctan.tug.org/tex-archive/macros/latex/contrib/texpower>.

- [21] Timothy Van Zandt. PStricks: PostScript Macros for Generic T_EX, March 1993. <http://ctan.tug.org/tex-archive/graphics/pstricks/obsolete/doc>.
- [22] Rolf Niepraschk. Ps4pdf Package, November 2004. <http://ctan.tug.org/tex-archive/macros/latex/contrib/ps4pdf>.