

定番プレゼンソフトを越えて: 聴衆に配慮したスライドを L^AT_EX とスクリプト言語で作成した例の報告

Beyond standard slideware: Audience-oriented slide preparation using L^AT_EX and scripting language

轟 眞市 Shin-ichi Todoroki

National Institute for Materials Science,
Namiki 1-1, Tsukuba, Ibaraki 305-0044, Japan
Todoroki.Shin-ichi@nims.go.jp

KEYWORDS Rule of Three, slides, presentation, handout, *Kata*, *Shu-Ha-Ri*

ABSTRACT 多くの人々は、事前に発表内容全体を見据えてコアとなるメッセージを特定すること無しにスライド作成に着手してしまうので、結局本番の発表で聴衆の注目を集め損ねてしまう。この過ちを犯さない様にするために、筆者は“Rule of Three”をスライド全体に適用することを心がけている。コアとなる3つのメッセージとそれに関連する語句をスライド全体に散りばめるために、L^AT_EX とスクリプト言語を使って自動化する環境を整えた。聴衆に配慮する姿勢を身につけるためのこの取り組みに際して、日本の伝統文化の継承にみられる「型」の考え方を意識した。

1 はじめに

プレゼンテーション・ソフトウェアを配慮なしに使って講演者の意図をぼかしてしまうことに対する警告が数多く発せられている [4, 6]。その中のひとつに、2008 年にベストセラーとなって 14ヶ国語に翻訳された本、「プレゼンテーション Zen」[8]がある。その著者は「準備の初期段階で最も大切なのは、コンピューターの前から離れ、て全体を見据え、コアとなるメッセージを特定することだ、と説いている。

筆者も別な方法で自分のメッセージを明確化するための時間を割いている—スライド全体に“Rule of Three”を適用するのである。この方針に基づいて、あらかじめ話す内容における3つのポイントを決め、それらに関係する語句をすべてのスライドに配置している。これに伴う編集作業を軽減するために、L^AT_EX とスクリプト言語のプログラミング機能を援用している。

図 1 に、筆者によるスライド作成例を示す [14]。簡潔なまとめを3行、最後のスライドに掲げており、講演の途中にもその一部を提示している (Slide 10 と 15)。残りの構造の説明は後述するが、この適切な冗長性を含んだ形式は、聴衆が話の展開に集中する助けとなっ

ている。

筆者がこの形式の基本的な考え方を学んだのは、1995年に受講したプレゼンテーションセミナーであった [3]。それ以降、この形式を L^AT_EX を使って発展させ、主に学術プレゼンテーションにおいて利用してきた。

筆者がこの形式でスライドを準備する際には、「型」[9]に則ることを意識している。「型」とは、古典芸能や武道をはじめとする日本の伝統文化の伝承過程に関わる基本的な概念である (図 1 の Slide 1 参照)。「型」を英訳すれば一般的には“form”だが、その語をもって日本語の「型」を完全に言い尽くすことはできない。

多くの日本の伝統技芸において、弟子は師匠が示す「型」を真似する稽古を積む。「型」そのものは芸ではなく、芸に到達するための道を示すものであり、機能性、合理性、美的性格が集約されている。

「型」を習得する段階を表す言葉に「守」があり、さらに「破」と「離」へと進む (図 2 参照)。「破」の段階においては、「型」から脱却するのだが、師匠の影響から逃れようとする気持ちがある分、独自の境地には至らない。そして「離」は、自分自身の「型」を確立する段階である。

1995年に受講したセミナーで学んだことを、筆者は「型」と認識した。おそらく、講師にはそのような意図は無かったであろう。以下の本論では、L^AT_EX で表現された筆者の現在の「型」、そのソースコードを管理する方法、そして「型」によるスライドがもたらした効果について述べる。

2 “Rule of Three” をスライド全体へ適用する「型」

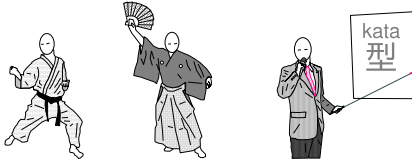
“Rule of Three” を講演全体に適用した場合の構造を図 3 に示す [2, 3]。この構造をもつ図 1 に示したスライドは、聴衆に話の流れを見失わせないための工夫を、2つの「型」を通して盛り込んでいる。

ひとつ目の「型」は、話の流れを確認するスライド (Slide 5, 10, 15, 19) を途中で配置することである。これらを本論文では「鳥の目スライド」と呼ぶことにする。Slide 5 では、Slide 19 と対の関係にある 3つの疑問文を提示し、講演の冒頭で聴衆の興味を引きつけることを狙っている。途中の Slide 10 と 15 では、直前に話したポイントのまとめを提示すると共に、次のポイントに対応する疑問文を再提示している。

ふたつ目の「型」は、「鳥の目スライド」を除くすべてのスライドの左上にヘッダを配置することである。ヘッダには、3つのポイントを示唆する短い語句を色を変えて配置し、聴衆への案内役を務めさせている。これらのスライドを、本論文では「蟻の目スライド」と呼ぶことにする。

これらの「型」を表現するためには、同じ語句をいろいろな場所に配置しなければならない。そこで、T_EX のマクロ定義を利用して、文字列の管理を一元化するのが良い。表 1 に、これらのスライドにおいて定義したマクロと、それを呼び出しているスライドの番号を示

プレゼンテーション資料作成作業への
Rule of Three 適用支援
物質・材料研究機構
光材料センター 轟 眞市



Slide 1

ソースの管理 テンプレートと Makefile の自動生成

```
$ head -1 'which seminit'
#!/usr/bin/ruby
$ seminit --help
usage: seminit [ --double ] [ --jp ]
$
```

- 単純三部形式、複合三部形式 (3×3)
- 英語、日本語
- latexmk と自作スクリプトを呼び出す Makefile

Slide 12

内容

プレゼン資料に Rule of Three を!

ソースの書法

必要最小限の命令は何か?

ソースの管理

どのように手間を省いているのか?

その波及効果

どんなご利益があるのか?

Slide 5

内容

プレゼン資料に Rule of Three を!

ソースの書法

定型文字列のマクロ定義、色指定、オーバーレイ指定

ソースの管理

用途に応じたテンプレートの自動生成、Makefile の整備

その波及効果

どんなご利益があるのか?

Slide 15

ソースの書法

オーバーレイ指定

内容 プレゼン資料に Rule of Three を! ↓ ソース

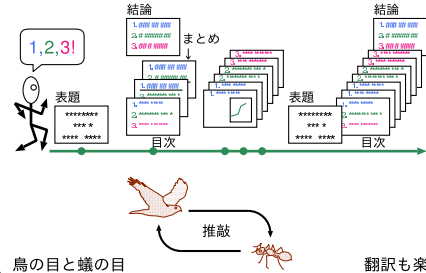
```
% prosper.cls の場合 (色指定等は省略)
\overlays{3}{
\begin{slide}{}
... \fbox{\TitleA}...
\fromSlide{2}{\fbox{\TitleB}...}
\fromSlide{3}{\fbox{\TitleC}...}
\end{slide}
}
```

↑ Slide 5

Slide 8

その波及効果

見通しの良い執筆プロセス



Slide 16

内容

プレゼン資料に Rule of Three を!

ソースの書法

定型文字列のマクロ定義、色指定、オーバーレイ指定

ソースの管理

どのように手間を省いているのか?

その波及効果

どんなご利益があるのか?

Slide 10

結論 プレゼン資料作成作業への Rule of Three 適用支援

ソースの書法

定型文字列のマクロ定義、色指定、オーバーレイ指定

ソースの管理

用途に応じたテンプレートの自動生成、Makefile の整備

その波及効果

執筆工程も配布資料も見通しが良い。外部の評判も上々

Slide 19

図 1. 講演 [14] で用いたスライド (抜粋)。

守破離

図 2. 守破離 — 川上不白 (1719–1807)。書: 土屋博之。

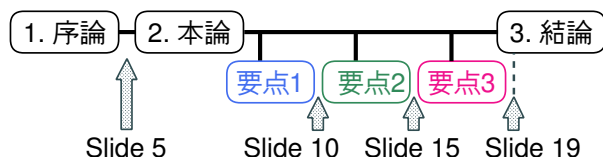


図 3. “Rule of Three” を講演全体に適用した場合の構造。スライド番号は図 1 に示したものに对应し、その構造の節目で話の流れを確認する役割を負っている。

表 1. 図 1 に示したスライドで定義されている定型文字列。*印の付いたスライドでは、当該文字列はオーバーレイ機能によって切り替え表示されているので、図 1 には表示されていない(図 4 参照)。

マクロ定義	呼び出しているスライドの番号
<code>\def\TitleOne{ソースの書法}</code>	5–10, 15, 19
<code>\def\IntroOne{必要最小限の命令は何か?}</code>	5, 10*
<code>\def\SummaryOne{定型文字列のマクロ定義、色指定、...}</code>	10, 15, 19
<code>\def\TitleTwo{ソースの管理}</code>	5, 10–15, 19
<code>\def\IntroTwo{どのように手間を省いているのか?}</code>	5, 10, 15*
<code>\def\SummaryTwo{用途に応じたテンプレートの自動生成、...}</code>	15, 19
<code>\def\TitleThree{その波及効果}</code>	5, 10, 15–19
<code>\def\IntroThree{どんなご利益があるのか?}</code>	5, 10, 15, 19*
<code>\def\SummaryThree{執筆工程も配布資料も見通しが良い。...}</code>	19

した。これらのマクロ群は、「全体を見据え、コアとなるメッセージを特定」して初めて定義できる。

なお「鳥の目スライド」では、話す内容に対応して表示する文字列を追加していくオーバーレイ機能を利用している。図 4 に示した 3 枚のスライドがその例であり、左下の表に示した様に、マクロ定義した文字列を切り替えて表示している。これにより、聴衆は講演の構造を思い出すことができるし、発表者は Slide 5 と 19 に提示した文章を暗記する必要がなくなる。

ところで筆者は普段、古典的な SEMINAR クラス [7] と PROSPER クラス [1] をスライド作成に使っている。その理由は単に、「型」を使い始めた 1995 年当時、私にとって最も魅力

Slide 10-1

Slide 10-2

Slide 10-3

呼び出されるマクロ	Slide 10		
	1	2	3
<code>\TitleOne</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>\IntroOne</code>	<input type="radio"/>		
<code>\SummaryOne</code>		<input type="radio"/>	<input type="radio"/>
<code>\TitleTwo</code>			<input type="radio"/>
<code>\IntroTwo</code>			<input type="radio"/>
<code>\SummaryTwo</code>			
<code>\TitleThree</code>			<input type="radio"/>
<code>\IntroThree</code>			<input type="radio"/>
<code>\SummaryThree</code>			

図 4. 「鳥の目スライド」におけるオーバーレイの効果 (表 1 参照)。

的に思えたものが SEMINAR パッケージだったからであり、その後これに対する上位互換性をほぼ有している PROSPER パッケージが登場したからである。Slide 10 のオーバーレイ機能を記述した PROSPER コードを、付録の図 8 に示す。

3 スクリプト言語の併用によるソース管理

前節で述べた「型」を表現するために、 \LaTeX のソースファイルを毎回書き下ろすのは大変に手間がかかる。そこで、必要なマクロ定義やコマンドを記述したテンプレートを用意しておくのが現実的である。しかし、講演する言語や講演時間の長短に応じてテンプレートの数は増えていく。例えば筆者の場合、30 分以上の講演では、3 つのポイントを含む単位を 3 つ続ける複合三部形式を採用している (そのスライド作成例は [13] 参照)。これらを一元的に管理するために、テンプレートファイルを自動生成する Ruby スクリプトを自作して活用しており (図 1 の Slide 12 参照)、その詳細は付録で解説する。単一のソースコードから PROSPER ベースのスライドと SEMINAR ベースの配布資料を生成させる方法も述べる。

4 波及効果

筆者がこの「型」に則ったスライドを 14 年間使ってきた経験から見えてきた利点を 3 つ挙げておく。

ひとつは、スライドの執筆が以前よりも見通し良く進められるようになった点である。「鳥の目スライド」を作成しないと「蟻の目スライド」に取りかかれぬので、常に部分と全体を考えて執筆する様になった(図 1 の Slide 16 参照)。科学技術プレゼンテーションの場合、発表前に追加の実験データを取ることは日常茶飯事である¹。発表申し込みの段階で立てた方針に従って「鳥の目スライド」を作成しても、その後の展開で軌道修正を迫られる場合も多々遭遇する。そんな時でも表 1 の様に文字列を一元管理しておけば、最小の手間で全体に変更を加えることができる。

マクロの一元管理をしておくと、スライドの翻訳も容易である²。講演で使う言語を解さない聴衆がわづかでも参加している場合に、翻訳したスライドを別に上映したり [15]、縮刷版を手渡しておくといった対応が可能になる。

ふたつ目は、スライドの縮刷版(紙にスライドを 4~6 枚印刷したもの)の見通しも良くなることである。縮刷版を事前に聴衆に配布するのは絶対に避けるのがプレゼンテーションのセオリーであるが [8]、やむを得ない事情でそうせざるを得ないことも多い。しかしながら、いつ縮刷版を配ろうとも、それに見通しの良さが備わっていなければ、結局その資料は読んでもらえないことになる。

ここで注意すべきは、スライドの縮刷版とポスターを同一視してはいけないことである。ポスターにはポスター用の構成法があり、それに則ったポスターに比べれば、会場を歩き回っている聴衆の注目を集めにくい³。

みつ目は、筆者が行ってきたプレゼン技術向上の啓蒙活動に対する評判が良いことである。2000 年以降、「型」を用いたスライド作成法を説く記事の執筆 [10, 12] や講義を続けてきた(ただし L^AT_EX の使い方には触れていない。講演 [14] の Slide 18 参照)。最も興味深かった反響を紹介する。ある理科系大学で筆者は 2 年生に対する講義(年 1 回、80 分)を担当している。「型」に則ったスライドを使って講義をしているので、彼らにしてみれば、目と耳から同時に学んでいることになる。やがて彼らが研究室に配属される頃から、学生が学会発表で賞をもらってくるが多くなったそうだ。これは、「守」の段階を示す好例なのかもしれない。

1. このため多くの研究者は発表直前になって初めてスライド作成に着手する。時間不足が仇となり、コアとなるメッセージがはっきりしないスライドが量産されてしまうのだと筆者は考える。

2. スライド内に 2ヶ国語を併記するのは避けるべきである。ごちゃごちゃして読みにくくするだけである。図 1 に示したスライドを翻訳したものが文献 [5] である。

3. 本論文で紹介した「型」をポスターに適用した例を挙げておく [11, 16]。

5 おわりに

L^AT_EX とスクリプト言語を使い、本文で述べた 2 つの「型」を通じて “Rule of Three” をスライド全体に適用することで、聴衆に配慮したスライドを作成した。この方針は、執筆者にスライドの執筆前にコアとなるメッセージを明確化させるとを強いる効果があり、また「型」の表現を通して執筆者にも聴衆にも講演全体を鳥瞰する視点を植え付ける。他の定番とされているプレゼンテーション・ソフトウェアにも「型」の表現を容易にするプログラミング機能が搭載されることを期待している。

謝辞

筆者の「型」に至る基本的な考え方をご教示頂いた Charles LeBeau 氏と Michael Strother 氏、および、筆者の求めに応じて快く書をしたためてくれた土屋博之博士に心より謝意を表します。

付録 A 付録

“Rule-of-Three” を適用したスライドを生成する最小限のコードを、筆者のホームページで公開している⁴。配布物には 2 つの Ruby スクリプト、2 つのスタイルファイル、そしてひとつの T_EX ファイルが含まれており、筆者の環境である Debian GNU/Linux 5.0.3 (lenny) でテストしたものである。この節ではこれらのファイルの役割を簡単に記しておく。

A.1 seminit と SEMTdrk.sty

seminit はテンプレートファイルと Makefile を生成するコマンドである。以下に示した 6 つのコマンドが、スライド作成に必要な最低限の工程である。

```
- mkdir foo
- cd foo
- seminit          # テンプレートと Makefile の生成
- make edit       # テンプレートの編集
- make            # スライド用 PDF ファイルの作成
- make sem4view   # 配布資料用 PDF ファイルの作成
```

この例では、seminit は *foo.tex* という名前の SEMINAR クラスを使ったテンプレートファイルを生成する (図 5 の上の囲みを参照)。seminit には 2 つのコマンドラインオブ

4. http://www.geocities.jp/tokyo_1406/slides4ajt.html

ションが指定でき、2種類ある三部形式(単純と複合)のうちのどちらかと、和英どちらの言語かが選択できる(図1のSlide 12参照)。このファイルでは、PROSPER固有のコマンドも記述できるが、SEMTdrk.styに記述された命令で無効化される(図6参照)。

この段階でプレビューできるイメージは、SEMINARクラスのarticleオプションを使ったものである。このイメージは配布資料を作るのにも使われる。make sem4view実行時にpsnupを起動して縮刷版を作成する。

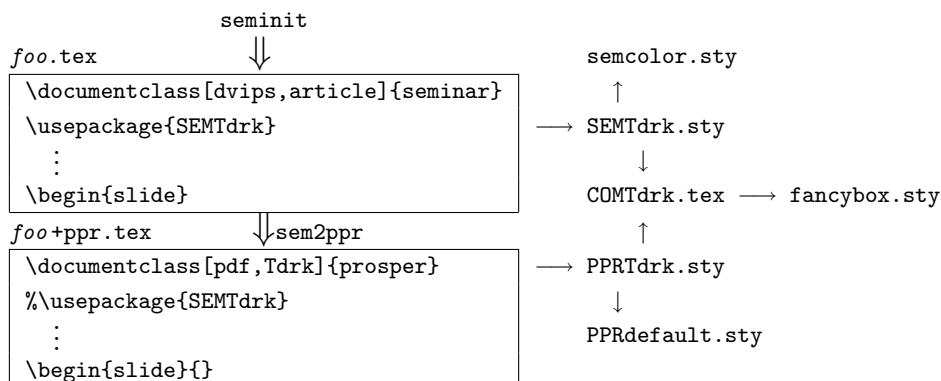


図5. sem2pprによるソースコード変換(左)と、関係するスタイルファイルの依存関係(右)。

```

1  \newcommand{\overlays}[2]{#2}
2
3  \long\def\fromSlide{%
4    \@ifstar\fromSlide@E\fromSlide@NE}
5  \long\def\fromSlide@NE#1#2{#2}
6  \long\def\fromSlide@E#1#2{}
7
8  \long\def\onlySlide{%
9    \@ifstar\onlySlide@E\onlySlide@NE}
10 \long\def\onlySlide@NE#1#2{#2}
11 \long\def\onlySlide@E#1#2{}
12
13 \long\def\untilSlide{%
14   \@ifstar\untilSlide@E\untilSlide@NE}
15 \long\def\untilSlide@NE#1#2{#2}
16 \long\def\untilSlide@E#1#2{}
17
18 \newenvironment{itemstep}{\begin{itemize}}{\end{itemize}}

```

図6. SEMTdrk.sty内に記述された、PROSPER固有命令の無効化。

A.2 sem2ppr と PPRTdrk.sty

PDF プレビューアを使って PROSPER 固有コマンドの効果を確かめる場合に `make` を実行するのだが、この時に呼び出されるのが `sem2ppr` である。このスクリプトは、図 5 に記した 3 種類の命令を書き換えて PROSPER クラスのソースコードに変換された `foo+ppr.tex` という名前のファイルを生成する。

その結果、変換後のソースは `SEMTdrk.sty` の代わりに `PPRTdrk.sty` を読み込む様になり、すべての `slide` 環境に空の引数が付加される (図 5 の下の囲みを参照)。`PPRTdrk.sty` の中では `PPRdefault.sty` が呼び出された後、2 つのクラスのレイアウト差を解消するために命令の再定義が行われる。

A.3 COMTdrk.tex

このファイルは 2 つのスタイルファイルのどちらからも読み込まれるものであり、ヘッダとタイトルを配置する `\pagetop` の定義が記述されている (図 7 参照)。PROSPER クラスにおける `slide` 環境の引数を空にしているのも、その代用として用いる命令である。

```

1 \usepackage{fancybox}
2
3 \newcommand{\myframe}[1]{%
4   \shadowbox{\small #1}\hfill\llap{\rule{0.95\textwidth}{1pt}}
5 \newcommand{\pagetop}[3]{%
6   \textcolor{#1}{\myframe{#2}}\llap{\raisebox{1ex}{\large #3}}%
7   \par\vspace{-0.5\slideparskip}}

```

図 7. `COMTdrk.tex` 内に記述された `\pagetop` の定義。

A.4 「鳥の目スライド」と「蟻の目スライド」

図 8 と 9 に示したソースコードは、それぞれ「鳥の目スライド」と「蟻の目スライド」のひとつであり、ここからオーバーレイ効果が埋め込まれた Slide 10 と 16 が生成される。

参考文献

1. Frédéric Goualard and Peter Møller Neergaard, *prospcr*, CTAN:macros/latex/contrib/prospcr/.
2. David Harrington and Charles LeBeau, *Speaking of speech: Basic presentation skills for beginners*, new ed., Macmillan Education, September 2008 (初版は 1996 年発行).
3. Charles LeBeau and Michael Strother, *Technical presentations*, One of the hands-on seminars titled “Power Presentation” held by NIC Tokyo, March 1995.
4. Peter Norvig, 青木 靖 (訳), PowerPoint の罫, http://www.aoky.net/articles/peter_norvig/lancet.htm.

```

1 \overlays{3}{
2   \begin{slide}%{} sem2ppr が自動的に付加するので不要
3   \begin{center}\doublebox{内容}\end{center}
4   {\small\begin{center}プレゼン資料に Rule of Three を!\end{center}}
5
6   \textcolor{blue}{\shadowbox{\TitleOne}}
7   \newline\hfill
8   \onlySlide*{1}{\slshape\IntroOne}
9   \fromSlide{2}{\SummaryOne}
10
11  \onlySlide{3}{
12    \par\smallskip\rule{\textwidth}{1pt}\par\smallskip\hfil\hfil
13    \begin{minipage}{0.75\textwidth}
14      \pagetop{green}{\TitleTwo}{}\hfill{\slshape\IntroTwo}
15    \end{minipage}\par\bigskip
16    \hfill
17    \begin{minipage}{0.75\textwidth}
18      \pagetop{red}{\TitleThree}{}\hfill
19        \textcolor{gray}{\slshape\IntroThree}
20    \end{minipage}\par\bigskip
21  }
22 \end{slide}
23 }
```

図 8. 「鳥の目スライド」である Slide 10 (図 4 参照) のソースコード。14 行目と 18 行目に \pagetop を使っている (図 7 参照)。

5. Shin-ichi Todoroki, *Upgrade your slides with Rule of Three and L^AT_EX*, T_EX Conference Japan 2009, August 2009, <http://pubman.mpg.de/pubman/item/escidoc:94908> (講演 [14] に用いたスライドの英訳).
6. Edward Tufte, *PowerPoint is evil*, Wired Magazine, September 2003, <http://www.wired.com/wired/archive/11.09/ppt2.html>.
7. Timothy Van Zandt, *seminar*, CTAN:macros/latex/contrib/seminar/.
8. ガー・レイノルズ, 熊谷 小百合 (訳), プレゼンテーション Zen, ピアソンエデュケーション, September 2009.
9. 源 了圓, 型と日本文化, pp. 5–68, 創文社, 東京, July 1992.
10. 轟 眞市, プレゼンテーションと情報管理, April 2004, <http://pubman.mpg.de/pubman/item/escidoc:28491>.
11. _____, ファイバビューズの自然停止に伴う 2 重周期的空孔列の生成, 日本セラミックス協会 2006 年年会講演予稿集 (東京都目黒区), March 2006, (2P134), p. 220, <http://pubman.mpg.de/pubman/item/escidoc:28396>.
12. _____, セレンディピティを高めるプレゼンテーション技術 (連載全 6 回), 工業材料 55 (2007 ~ 2008), 8 ~ 翌 3 月号, <http://pubman.mpg.de/pubman/item/escidoc:28426>.
13. _____, だからセルフアーカイビングはやめられない!, 第 1 回 SPARC Japan セミナー 2009 「研究者は発信する - 多様な情報手段を用い、社会への拡がりを求めて」, June 2009,

```
1 \overlays{2}{  
2 \begin{slide}%{} sem2ppr が自動的に付加するので不要  
3 \pagetop{red}{\TitleThree}{見通しの良い執筆プロセス}  
4 \includegraphics[width=\textwidth]{schedule}\par\medskip  
5 \fromSlide{2}{  
6 鳥の目と蟻の目\includegraphics[width=0.4\textwidth]{bird-ant}  
7 \hfill 翻訳も楽  
8 }  
9 \end{slide}  
10 }
```

図 9. 「蟻の目スライド」である Slide 16 (図 1 参照) のソースコード。

<http://www.nii.ac.jp/sparc/event/2009/20090625.html>.

14. _____, プレゼンテーション資料作成作業への Rule of Three 適用支援, TeX ユーザの集い 2009 予稿集 (東京), August 2009, p. 11, <http://oku.edu.mie-u.ac.jp/texconf09/proceedings.pdf>.
スライドの入手先: <http://pubman.mpg.de/pubman/item/escidoc:94877>.
15. _____, プレゼン修行拾遺録 (2) バイリンガル・プレゼンテーション, マテリアルインテグレーション 22 (2009), no. 2, 57-58, <http://pubman.mpg.de/pubman/item/escidoc:61282>.
16. 轟 眞市, 井上 悟, ファイバヒューズ点火現象の直接観察, 第 46 回ガラスおよびフォトニクス材料討論会講演要旨集 (滋賀県彦根市), November 2005, (P06), pp. 42-43, <http://pubman.mpg.de/pubman/item/escidoc:28394>.