

Typography and Desktop Publishing

Andreas Rüping

www.rueping.info

e-mail: rueping@acm.org

Introduction

Desktop publishing software offers a variety of features to influence the typography and the layout of documents[3][4]. However, typography is a science and an art in itself. To produce documents that look good and read well, authors must use these features with care.

This paper presents a few aspects of typography in pattern form. The paper serves the following purposes:

- It describes and motivates five typographical rules that are quite important and that can easily be implemented on standard desktop publishing systems. Users of such systems can apply these rules immediately to their own documents.
- It demonstrates that typographical rules aren't arbitrary rules, but are in fact patterns. Each rule is a solution to a problem and resolves a number of forces, even if readers normally aren't aware of the effects that these forces represent.

The patterns in this collection apply to technical documents printed on paper. Technical documentation for software forms the background of this collection of patterns. Still, the patterns also apply in non-software domains. However, documents are beyond the scope of this collection of patterns if they are not going to be printed, such as online documentation or overhead presentations.

Naturally, the collection of patterns in this paper is far from complete. Interested readers are invited to search the rich body of literature on typography for further typographical patterns.

Guidelines for the Readers

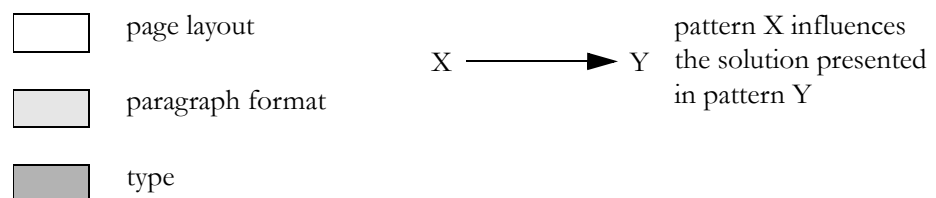
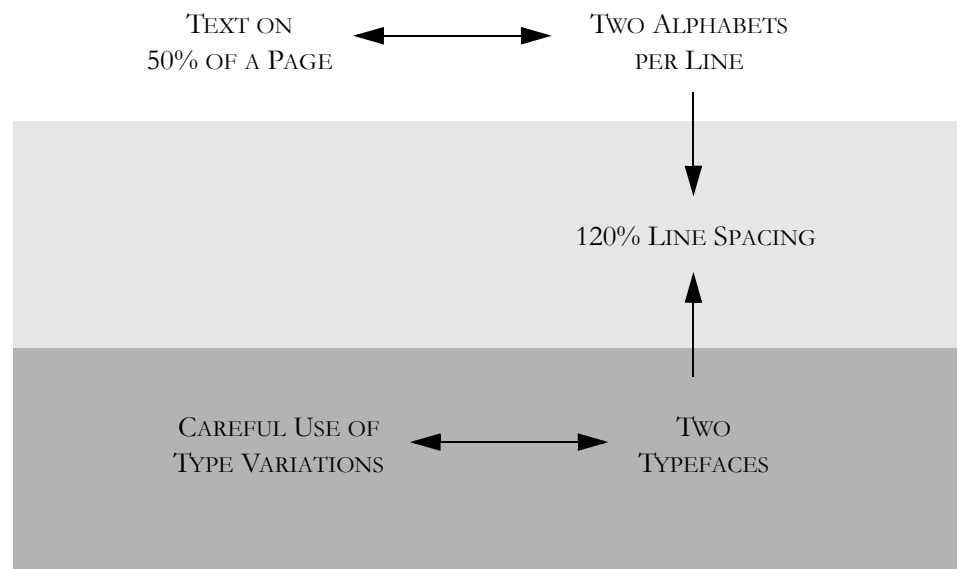
This paper is part of a larger effort describing issues in technical documentation in pattern form. In particular, this paper builds upon the collection of patterns on the structure and layout of technical documents [5]. That collection of patterns includes a pattern called LAYOUT AND TYPOGRAPHY which states that typographical rules are important and gives a few brief examples. The patterns in this paper refine the LAYOUT AND TYPOGRAPHY pattern and present some principles of typography in more detail.

The pattern form used in this paper is the same that is used in the other pattern papers on technical documentation. Each pattern starts with the problem section asking a question. The forces section describes the driving forces behind possible solutions. The solution section presents an answer to the question from the problem sections that resolves the forces as well as possible. The discussion section gives additional information and describes relationships to other patterns.

Each pattern provides a pattlet printed in boldface — the problem section and the first paragraph of the solution section. They form a thumbnail that gives an overall impression of the actual pattern.

The following roadmap diagram gives an overview.

Overview



1 Text on 50% of a Page

Problem

How much space on a page should be devoted to text?

Forces

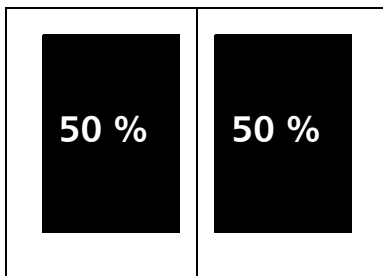
Page layout should be aesthetic in order to please the readers. Almost all readers prefer pages with ample margins over pages that appear to be crowded with text [7]. Margins are also necessary for functional reasons. The inner margin (also called the gutter) must allow enough space for binding. All margins must allow enough space for readers to hold a page without obscuring any text [7].

Margins that are too large, however, aren't appropriate for economical reasons since more paper than necessary is used. Extremely large margins can also create the impression that the author tried to unduly extend the document, and that the document's contents are rather light-weight compared to its length.

The optical centre of a page is the place first focused by the reader's eye. The optical centre is a little bit above the mathematical centre. This has an additional effect on the optimum margin sizes [1][7].

Solution

About 50% of the page should be devoted to text [1][6][7].



- Headers and footers do not count when the live area (the space enclosed by the margins) is calculated.
- In order to put the centre of the text near the optical centre, the text should be positioned a bit nearer to the top of the page than to the bottom of the page. Books require the margin size to increase from inner to top to outer to bottom margin. A ratio of 2:3:4:5 between the margin sizes is often recommended [3].
- The minimum gutter margin is 2 cm to allow binding.
- A live area covering slightly more than 50% of the page is acceptable when not all of the live area is actually covered by text, for instance due to the use of sideheads (headings that are placed to the left or the right of the actual paragraphs).

For instance, a standard A4 page has a size of 21×29.7 cm. Here margins of 2, 3, 4, and 5 cm meet the rule, leaving a live area of 15×21.7 cm. The live area space is 325.5 square cm, which is 52% of the A4 page.

Discussion

The 50% rule is surprising to many people at first. When people look at a printed page, they often overestimate the amount of space devoted to the live area and underestimate the amount of space devoted to margins. In fact, average readers estimate that the live area covers about 75% of a page when it does in fact cover only 50% [6]. In other words, 50% text is more than it seems.

There is a limit on the line width, saying that there should be about TWO ALPHABETS PER LINE. If a line across the live area of the page can contain significantly more than two and a half lowercase alphabets, techniques must be employed to reduce the line width. Having two columns per page is one option, using sideheads is another.

2 Two Alphabets per Line

Problem What is the optimum line width?

Forces

When reading, the reader's eyes travel along the line. The eyes make small, jerky movements called saccades in between which there are periods called fixations. Fixations last about 0.25 seconds while saccades are only 0.01 seconds long. It is during the fixations that information is picked up [2].

A line break interrupts the eye movement along the line. The reader's eyes have to shift back to the beginning of the next line. Short lines increase the number of line breaks. If lines are too short, the reader's eyes have to find the beginning of the next line more often than necessary, which breaks the flow of reading and makes reading tiresome [1][3].

On the other hand, lines that are too long also make reading a difficult and tiring job. Long lines make it difficult for the reader's eyes to follow a line and to find the beginning of the next line once a line break occurs [1][3].

Moreover, the optimum line width depends on the type and the type size used. Types of larger sizes require larger line widths [1][3].

Solution **One line should contain approximately two lowercase alphabets of the standard typeface.**

As a rule of thumb, one and a half lowercase alphabets are considered the lower limit of characters that should fit into one line. The upper limit lies near two and a half, at most three lowercase alphabets [3].

abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz abcdefghijk

If lines are too long, there are several ways to fix this problem:

- The author can choose a larger typeface or a larger type size.
- The author can make the lines shorter by increasing the margins.
- The author can make the lines shorter by using two columns rather than one.

Discussion

There is a subtle effect that spacing has on the range of acceptable line widths. When the standard of 120% LINE SPACING is slightly increased, line widths a little above two and a half lowercase alphabets can be acceptable.

When authors choose to optimise the line width either by increasing the margins or by using two columns, they should make sure that the page layout conforms to the TEXT ON 50% OF A PAGE rule.

Justified text can be problematic when the line width is near the lower limit. Since justification requires to adapt the spacing between words and maybe between characters, unnaturally long word distances can occur. It is therefore important to use hyphenation [7]. Using ragged right rather than justification can also be worth considering.

A slightly increased line width is acceptable for documents that readers concentrate on, as opposed to books that readers often hold in their hands and read much more casually. Line widths around two and half lowercase alphabets are therefore acceptable for most technical documents.

3 120% Line Spacing

Problem What is the optimum line spacing?

Forces

An even texture is crucial for the legibility of a document [7]. Reasonable spacing between words and lines is a prerequisite for an even texture. While word spacing is to a large degree determined by the typeface used, line spacing is not.

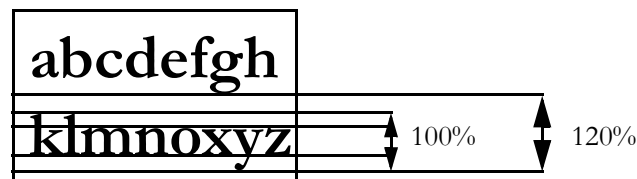
If there is too much line spacing, consecutive lines no longer form a unit, which they should, but appear to be separated from each other, which makes text difficult to read [7].

However, line spacing shouldn't be too small, either. A certain amount of leading is necessary to ensure that the ascenders of one line aren't jammed into the descenders of the previous line. (Leading is the space between the lines; the line spacing results from adding the type size to the leading.)

Next, the x-height of a typeface (the height of the lowercase letter x) influences its appearance. A typeface with a relatively small x-height appears to be smaller than its size suggests, and leaves more natural space between lines, thus reducing the need for extra leading.

Solution The best line spacing is roughly 120% of the type size.

In other words, 20% leading is normally fine. For standard type sizes such as 10, 11, or 12 point this means that 2 point leading is appropriate.



In the following cases the spacing may need some fine-tuning:

- Spacing can be decreased for typefaces with a relatively small x-height.
- Spacing can be increased for typefaces with a relatively large x-height.
- Spacing should be increased for long lines.

Discussion

120% line spacing is appropriate for standard text. Headings, however, can be an exception since they make the structure of a text visible. To help readers perceive the structure of a text, headings should stand out not only by an increased type size, but also by a line spacing that may exceed 120%.

Increasing the spacing for long lines applies in those cases where lines are significantly longer than the optimum line width defined by the TWO ALPHABETS PER LINE pattern. However, even with increased spacing lines are inappropriately long when they contain significantly more than two and a half lowercase alphabets.

While spacing is one technique to make the structure of a text visible, using different typefaces and sizes for headings and for body text is another and closely related technique (see TWO TYPEFACES).

4 Two Typefaces

Problem How many typefaces should authors use, and which?

Forces

Authors can use different types and different type sizes to express the different meanings of text, such as headings, emphasis, references, citation, and do on.

Often different types are not necessary, though, since different type variations can express the same different meanings equally well, or even better.

In fact, using a large number of different types is problematic for aesthetic reasons as well as for legibility reasons. Many different types give a document a chaotic appearance which is generally unappreciated by the readers.

Typefaces can express things such as soundness, formality, innovation, fashion, and so on [1]. Typefaces should be chosen with respect to what they represent. Only typefaces that fit together should be mixed.

Serif typefaces are more legible than sans-serif typefaces. (Serifs are the short lines that cross the ends of the strokes of a printed letter.) However, single pieces of text printed in a sans-serif typeface stand out from the main text and can attract the reader's eye [3].

Solution

In most cases, two typefaces per document are appropriate — one for the body text and one for the headings. A serif typeface should be chosen for the body text, while for the headings a sans-serif typeface is generally fine.

- There is nothing completely wrong with using only one typeface over an entire document. In this case a serif typeface should be chosen for legibility reasons. However, a second typeface can polish up a document's appearance.
- Using more than two different typefaces is almost always inappropriate. A possible exception is the use of a third typeface for code fragments included in a document; still, the third typeface should be used sparingly.
- The type size for body text should be 10 to 12 point; 14 to 18 point is appropriate for headings.

When two different typefaces are used, typefaces should be chosen that are not too similar, so that they can easily be told apart, but that still fit well together in an aesthetic sense. Here are two examples [3].

Example

Here Frutiger is the typeface chosen for the heading. The body text is printed in Garamond.

Example

Here Helvetica is the typeface chosen for the heading. The body text is printed in Times New Roman.

Discussion

The type sizes mentioned apply only to printed documents. Online documents require different type sizes, but are beyond the scope of this pattern.

In particular, there is absolutely no need to express emphasis through different type faces. In fact, it's counter-intuitive. All necessary kinds of emphasis can be expressed by a CAREFUL USE OF TYPE VARIATIONS.

5 Careful Use of Type Variations

Problem

How can parts of a text be emphasised?

Forces

Authors can employ type variations to express emphasis, importance, cross-references, etc. When used this way, different type variations are helpful for readers to understand the text, and in particular to understand the particular role that some words in a text take on.

Normal lowercase words appear in a characteristic shape defined by the ascenders and descenders of the letters contained. A characteristic shape is crucial for a word's legibility. Many type variations don't feature the characteristic shape as much as a standard lowercase typeface does and therefore decrease the legibility [6].

Words printed in italics still have a characteristic shape. Nonetheless, italics slightly decrease the legibility of text. Reading a text printed in italics takes about 4% more time than reading the same text printed in a standard lowercase type [6].

All capital letters do not feature a characteristic shape at all. They decrease the legibility of text quite dramatically. A text printed in all capital letters is read about 12% more slowly than a normal text [6]. Moreover, all caps are not appreciated by a vast majority of readers.

Underlines and all capital letters break the flow of a text. Underlines are a common technique only on typewriters, where no other style elements are available. Both underlines and all capital letters are hardly ever used in printed books.

Solution

Authors should use type variations with care [1].

The following type variations are considered fine style elements:

- Boldface can be used to emphasise single paragraphs.
- Italics are commonly used to place emphasis on a particular word.
- Small caps are often used to represent cross-references or people's names.

All capital letters and underlines should not be used.

shape

shape

SHAPE

Discussion

Using underlines is fairly common for hyperlinks in online documents. There may be some justification for this, however, underlines should be avoided in documents that are printed on paper. Since printed documents are the scope of this paper, the exclusion of underlines stands.

Conclusions

The literature on typography does not only present typographic rules and techniques, but also emphasises the role of creativity [7]. Creativity allows to define well-crafted layouts that make documents attractive to the readers.

The appendix presents two examples that are quite different; still, both demonstrate how the patterns in this paper can be applied in creative ways. The first example is a page from a book very well known in the patterns community, *Design Patterns* by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. There is ample room left free on the page and the spacing chosen adds to the book's well-balanced appearance. One serif type face is used for the body text, and a different serif type face is used for the headings. A sans-serif type face is reserved for the diagrams. Type variations are used carefully; the exemplary page contains just one word printed in italics for emphasis.

The second example is not taken from a book, but is an excerpt from a technical documentation that was produced with *javadoc*, the documentation generator that is part of the Java Developers Kit. Although its appearance is quite different from that of a book, the same patterns are at work here to make the document well readable.

As the examples show, the patterns presented in this paper leave ample room for creativity. They provide a framework for increasing the legibility and the aesthetics of printed documents. They can be applied in many different ways, and leave a lot of their implementation open to those who apply them. in a, hopefully, creative way.

Acknowledgements

Thanks are due to Charles Weir who, as the EuroPLoP shepherd for this paper, made many helpful comments and suggestions for improvement, and in particular suggested the inclusion of examples in the appendix.

Thanks are also due to John Vlissides and Addison-Wesley for providing an exemplary page from *Design Patterns*.

Last not least, thanks are due to the participants of the workshop at EuroPLoP '99 at which this paper was discussed.

References

- [1] Theodore E. Conover. *Graphic communications today*. West Publishing Company, 1985.
- [2] Robert G. Crowder. *The psychology of reading*. Oxford University Press, 1982.
- [3] Jürgen Gulbins, Christine Kahrman. *Mut zur Typography*. Springer-Verlag, 1992 (German language).
- [4] Donald E. Knuth. *Digital Typography*. Cambridge University Press, 1999.
- [5] Andreas Rüping. "The structure and layout of technical documents", in: Jens Coldewey, Paul Dyson (Eds.), *Proceedings of the Third European Conference on Pattern Languages of Programming and Computing*, Universitätsverlag Konstanz, 1998.
- [6] Miles A. Tinker. *Legibility of print*. Iowa State University Press, 1963.
- [7] Suzanne West. *Working with style: Traditional and modern approaches to layout and typography*. Watson Guptill, 1990.

Appendix A

A page from *Design Patterns* by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, published by Addison Wesley, 1995.

COMPOSITE 163

COMPOSITE

Object Structural

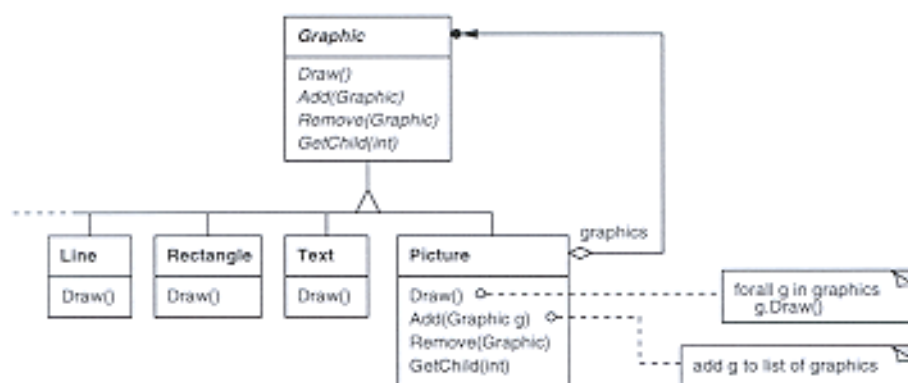
Intent

Compose objects into tree structures to represent part-whole hierarchies. Composite lets clients treat individual objects and compositions of objects uniformly.

Motivation

Graphics applications like drawing editors and schematic capture systems let users build complex diagrams out of simple components. The user can group components to form larger components, which in turn can be grouped to form still larger components. A simple implementation could define classes for graphical primitives such as Text and Lines plus other classes that act as containers for these primitives.

But there's a problem with this approach: Code that uses these classes must treat primitive and container objects differently, even if most of the time the user treats them identically. Having to distinguish these objects makes the application more complex. The Composite pattern describes how to use recursive composition so that clients don't have to make this distinction.



The key to the Composite pattern is an abstract class that represents *both* primitives and their containers. For the graphics system, this class is `Graphic`. `Graphic` declares operations like `Draw` that are specific to graphical objects. It also declares operations that all composite objects share, such as operations for accessing and managing its children.

Appendix B

An excerpt from a Java program printed by *javadoc*, a program included in the Java Developers Kit that generates HTML from Java sources.

Class Whiteboard

All Known Subclasses:

SimpleWhiteboard, CompositeWhiteboard

```
public class Whiteboard
extends java.lang.Object
```

Whiteboard represents the base class of both simple and composite whiteboards supported by the framework.

Method Summary

<code>java.lang.String</code>	<code>getName()</code> Returns the name of this Whiteboard object.
<code>void</code>	<code>setName(String)</code> Return a comment for this Whiteboard object.
<code>void</code>	<code>paint(Graphics)</code> Paints Whiteboard object.
<code>void</code>	<code>update(Graphics)</code> Updates Whiteboard object.

Method Detail

getName

```
public String getName()
```

Returns the name of the Whiteboard object.

Returns:

a string that contains the name of the Whiteboard object.

setName

```
public void setName(java.lang.String name)
```

Sets the name of the Whiteboard object.

Parameters:

name - name of the Whiteboard object.
