Document Production: Visual or Logical?

Leslie Lamport

24 February 1987

The Choice

Document production systems convert the user's input—his keystrokes and mouse clicks—into a printed document. There are many different ways of classifying these systems. I will discuss a classification based on the extent to which the user regards his document as a visual structure rather than a logical one. A system in which the user specifies a visual description of the output will be called a *visual* system, and one in which he specifies the logical structure of his document will be called a *logical* system. Visual systems may be more convenient for short, simple documents like love letters or laundry lists. However, I will argue that logical systems are better for more complex documents like books and technical articles.

In a purely visual system, one would simply paint a collection of pixels on the screen. The word *cat* would be no different from a picture of a cat–the user could change the shape of the *t* as easily as he could change the shape of the tail in a picture of a cat. Finding all instances of *cat* and replacing them by *dog* would be as hard as finding all cats in a picture and replacing them with dogs.

In a purely logical system, one would enter only the logical structure of a document, describing such things as words, paragraphs, theorems, sections, and cross-references. The system would translate this logical structure into a collection of dots on sheets of papers, with the user giving only general instructions—for example, specifying two-column output formatted for a conference proceedings.

There are no purely visual systems used for document production. All systems keep some logical representation of the document that they use to generate the pixels. The most primitive ones keep only the letters that generate the characters. In such a system one can easily find all instances of *cat*, but a search for all instances of *domestic* would miss the ones in which the word is hyphenated across lines. More sophisticated systems keep more of the logical structure, thereby acting more like logical systems. It is my thesis that such systems are good for serious document production only to the extent that they act like logical systems.

I know of no purely logical system that is currently available. Systems like *Scribe* and LaTeX permit the user to describe the visual appearance as well as the logical structure of the document—for example, by inserting a command to add a quarter-inch vertical space. The need to provide the user with such commands is a symptom of the deficiencies of these systems.

Current logical systems require the user to describe his document as a text string, filled with obscure-looking commands. This is a cumbersome way to represent the logical structure of a document; it is a sign of the primitive nature of these systems, not an inherent feature of logical document production. Systems can be built to allow more convenient editing of the document's logical structure. I'm not interested in the question of whether the inconvenience of describing the document with an ASCII text file is bad enough to make visual systems preferable. Choosing between two evils is never pleasant. I will confine myself to arguing the inherent superiority of logical systems to visual ones.

Computers Work Logically, Not Visually

In a recent paper, I used the notation f^e_x to denote the result of substituting e for x in f. With a visual system, I would have entered this notation by simply putting the e above and to the right of the f and the x below and to the left. Using LaTeX, I might have typed the formula as `f^{e}_{x}`, the `^` command indicating a superscript and `_` indicating a subscript. This input would have been a partially logical and partially visual description—logical because the subscript and superscript are denoted logically by commands rather than visually by placement, but visual because it describes the representation (super- and subscript) rather than the logical concept of substitution. I therefore chose to define a command `\subfor` of three arguments, and typed the formula f^e_x as `\subfor{f}{e}{x}`. The input then unambiguously describes the logical structure that it represents. For example, the input would distinguish the result of substituting 3 for i in s, represented as `\subfor{s}{3}{i}`, from the cube of the i^{th} element of a sequence s, represented as `s^{3}_{i}`, even though both are printed as s^3_i.

After I had completed the first draft of my paper, someone told me that I had used the wrong notation; the result of substituting e for x in f should be denoted by f_e^x rather than f_x^e. I had to reformat my paper to conform to the correct notation. Had I used a visual system, in which only the visual representation is maintained, I would have had to examine every page visually to find all instances of the notation and changed each one individually. Had I used the half-logical method, entering `f^{e}_{x}`, I could have written a program to find all text strings of the form `...^{...}_{...}` and allow me to choose whether to transform it. (Human intervention would still be required to prevent changing the cube of s_i to s_3^i.) Having chosen a logical representation, I merely had to change the definition of the `\subfor` command—a simple change at a single point in the text.

The ease of making the change when the notation was represented logically rather than visually was no fluke; it was a consequence of the fundamental fact that computers are good at processing logical information, but bad at processing visual information. Recognizing that f_x^e is a single logical entity with three components is a difficult problem of artificial intelligence if one is given only the visual representation, but it is a trivial programming exercise if f_x^e is represented as `\subfor{f}{e}{x}`.

Writing or Formatting?

The purpose of writing is to convey ideas to the reader. The worst aspect of visual systems is that they subvert the process of communicating ideas by encouraging the writer to concentrate on form rather than content. Ideas are conveyed by the logical structure of the text; the function of the visual format is to display this structure. The author should be concerned with the structure, not any particular visual representation.

Visual systems encourage the user to substitute formatting for good writing. A simple example is the use of vertical space. If there's an awkward transition from one paragraph to the next, the user of a visual system can simply add some vertical space between the paragraphs. But, what does this space accomplish? The awkward transition is still there; the reader is still jarred by it. The extra space simply declares that there is an awkward transition and the author is either too lazy or too bad a writer to fix it.

An awkward transition is a symptom of a poorly structured document; it can be fixed only by restructuring the document. A logical system forces the writer to think in terms of the document's logical structure; it doesn't give him the illusion that he is accomplishing anything with cosmetic formatting changes.

Phosphors or Ink?

Proponents of many visual systems boast that they let the user work with an exact replica of the printed page. In fact, a serious drawback of many visual systems is that they force the user to work with an exact replica of the printed page. When the author is editing his document, he becomes a reader. Like any reader, he wants to be presented with the document in a format that is easy to read. A format that is adapted to the printed page is a poor one for a screen. Phosphors are different from ink, and a screen is not a piece of paper; it is not easy to read a picture of a printed page on a screen.

A computer screen differs from a printed page in many ways, including resolution, width, and the availability of different colors. Each of these differences implies differences in the way information should be displayed. In addition to the differences in the two media, the presence of a computer behind the screen also has striking implications. Consider the problem of pagination. One of the worst features of books is the splitting of text across pages. It would be easier to read a document straight through, from front to back, if it were printed as a continuous scroll. We use books rather than scrolls because they are easier to produce and because documents are not always read in such a linear fashion. The computer offers the best of both worlds. We can scroll through text, avoiding distracting page breaks, and still move easily to another part of the document. It is senseless to use a computer to simulate a book, complete with page breaks.

A typical writer of technical material spends two to eight hours per page writing. He spends much of that time looking at the representation of the document on his screen. A visual system that forces the writer to view on his screen a version formatted for paper makes his task harder.

Who Should do the Formatting?

Logical systems attempt to remove formatting concerns from the author. The author specifies only the general form of the output—technical report, journal article, etc.—while the system makes the actual formatting decisions—amount of paragraph indentation, amount of space above a displayed equation, etc. Visual systems give free rein to the author's artistic tendencies, allowing him to format everything as he wishes. This would be fine if documents were meant to be displayed on walls and admired for their aesthetic qualities, but they're not.

The purpose of writing is to convey ideas to the reader. The purpose of formatting is to make the document easier to read, not to look pretty. Document design is a skill acquired through training and experience. A logical system can apply the skill of a trained designer to the formatting of a document. A visual system forces the author to do his own document design, often with disastrous results. Most authors are not competent designers and make typographic errors—formatting decisions that make the document harder to read.

A LaTeX user once complained because he wanted to format an equation to look something like this:

$$\forall i: \qquad f(x_i) > g(y_i) \tag{7}$$

Formatting the equation in this way would have been easy with a visual system; he would just have put everything where he wanted it. However, LaTeX provides no easy way to do this. The user just enters the equation and LaTeX formats it the way it wants. (It also assigns the equation number.) If the user declares the $\forall i$ to be part of the equation, the result looks like this:

$$\forall i : f(x_i) > g(y_i) \tag{8}$$

If he declares the $\forall i$ to come before the equation, then LaTeX makes it part of the text preceding the displayed equation.

This particular user found the formatting of (7) more aesthetically pleasing than that of (8), and I agreed with him. However, (7) is a typographical mistake. Equations are numbered so they can be refer to in the text. When the reader encounters a reference to (7), it is not immediately clear from the formatting whether it refers to the entire equation $\forall i : f(x_i) > g(y_i)$ or just to the inequality $f(x_i) > g(y_i)$. It is clear from the formatting that (8) refers to the whole equation and that, if the $\forall i$ were part of the preceding text, then the equation number would refer only to the inequality. The formatting of (7) introduces an ambiguity, making the document harder to read.

The purpose of document design is to display the logical structure of the document through its formatting, thereby making it easier to read. A user with no training in design is easily seduced by a visual system into formatting the document to be aesthetically pleasing, often making it harder to read.

A visual system can makes things hard even for a trained designer. An important principle of document design is uniformity—the same logical element should be formatted the same way throughout the document. It is difficult to achieve uniformity if the user must specify the formatting of each instance of the element. For example, all displayed quotations should be indented the same amount, but this is not likely to happen if the user must specify the amount of indentation whenever he types a quotation.

Must the User Ever Format?

There are two reasons why the author may have to specify formatting in a logical system. First, no logical system can provide a complete assortment of predefined logical structures. For example, a general-purpose system is unlikely to provide facilities for formatting recipes. The writer of a cookbook must tell the system how to format recipes—hopefully, after consulting a professional designer. A logical system should permit the user to define his own logical structures and to specify how they are to be formatted. Several different formats might have to be specified—for example, one for a single-column page, one for a double-column page, and one for the computer screen. In a logical system he does this once; in a visual system he must format each recipe individually.

The second reason for specifying formatting is to overcome an inherent problem with computers. Embodying design principles into programs is difficult, and a designer will always be able to do a better job of formatting an individual document than will a computer program that he devises. Achieving the highest possible quality requires the ability to make changes to the system's output. This will be a matter of fine tuning, changing such things as page breaks and figure placement. This is a visual process, and one would like a visual system for doing it—one that allows the user to manipulate screen images of the final output.

If such visual editing is ultimately desirable, why not use a visual system in the first place? The answer is that the flea should not wag the dog. The changes will generally be of such a minor nature that they are not worth bothering with in a preliminary version intended for a small audience, nor for any document that is not widely distributed. They will be done only when producing the final copy for the publisher.

Even using LaTeX, which does not make the final formatting very easy, I usually spend less than two minutes per page doing the final formatting to produce camera-ready output. This is insignificant compared with the two to eight hours per page I spend writing. There is much more to be gained by making writing easier than by simplifying the final formatting task.