International Journal of Business and Economics, 2003, Vol. 2, No. 2, 141-154

New Guidelines for Spreadsheets

John F. Raffensperger[*]
Deptartment of Management, *University of Canterbury*, *New Zealand*

Abstract

This article examines rules of style for text, graphics, and mathematics and applies these rules to spreadsheets. The article describes the new style in detail with an example and contrasts the new style with the existing programming style.

Key words: spreadsheets; readability; visual display

JEL classification: C60; M10

1. Is Writing a Spreadsheet Like Writing a Computer Program?

Proper writing of spreadsheets is a special concern to management educators and operations researchers. Educators should teach good habits and must read students' work. Operations research consultants use spreadsheets to produce math models for clients; these models can be large and complicated. Math modelling is now widely taught with spreadsheets, so teachers of math modelling must also teach spreadsheet use (see Winston, 1996, for example). While this article has an operations research point of view, it has general application.

Spreadsheets are often hard to read and often contain mistakes. Panko and Sprague (1998) found that around 2% of spreadsheet cells contain errors, implying that the likelihood of a correct model is less than 50% for a 35-cell spreadsheet. Why are spreadsheets so difficult? Part of the problem is due to existing prescriptions for spreadsheet style.

Most literature about spreadsheet style specifies a style based on computer programming. The computer program style is supposed to be a "standard" format with separate blocks for data, calculation, and output, as in Bromley (1985), Kee (1988), Kee and Mason (1988), and Stone and Black (1989). Some articles, such as Bissell (1986), Edge and Wilson (1990), and Crain and Fleenor (1989), mainly recommended a standard format with various mixes of heading, date, file name display, author, approval signoff line, table of contents, error summary box, instruction area, range names, global protection, absolute references where possible,

Received December 20, 2002, revised August 9, 2003, accepted September 16, 2003.
[*]Correspondence to: Department of Management, Private Bag 4800, University of Canterbury, Christchurch, New Zealand. E-mail: john.raffensperger@canterbury.ac.nz. Thanks to Linus Schrage and the gentle folk at Lindo Systems for their gracious help.

validation formulas for input and output, and no constants in formulas. While these seem like good ideas, they have not produced readable spreadsheets.

Most spreadsheet writers do not have a background in data processing (Cragg and King, 1993), so a “spreadsheet-as-program” metaphor is one that a reader is unlikely to expect, understand, or want. Cragg and King (1993) and Davis (1996) found that managers like spreadsheets for the freedom from internal information technology groups and may be hostile to perceived interference.

Edwards, Finlay, and Wilson (2000) wrote a good high-level overview of spreadsheet use, guidelines for planning spreadsheets, and “best practices” for verification. By contrast, this article focuses more on cell-by-cell construction and for the first time draws on literature about style for text, mathematics, and graphics.

Abstractly, it is true that a spreadsheet has three parts: data, calculation, and output. However, this functional structure was originally designed for the computer’s needs of batch-processed punched cards, rather than the reader’s needs. The input-calculation-output structure implies the calculation is a black box that the reader should ignore rather than read. Instead of “data and calculation,” the reader will be thinking in terms of the business problem, such as “person, shift assignment, preference.” This suggests that a spreadsheet should be organised to follow the business logic.

Conway and Ragsdale (1997) cited and commented on Kee (1988), “‘Reliable spreadsheet software begins with a standard format for developing spreadsheet applications....’ However, contrary to Kee’s sentiments (given above), for many optimisation problems we find that the forced use of a standard format results in spreadsheets models [sic] that are more difficult to construct, less reliable, and more difficult to understand.” Later: “In most cases, we believe the spreadsheet design which communicates its purpose most clearly will also be the most reliable, auditable and modifiable design.”

Conway and Ragsdale wrote the only paper with ideas other than “have modules for input, computation, and output, and avoid constants in formulas.” Their important new proposals are (1) that related formulas should be in physical proximity, and (2) how we should write depends on how we read.

This article significantly extends the emphasis on readability by examining how spreadsheet developers can apply styles for text, graphics, and mathematics to spreadsheets. A spreadsheet is a mixture of text, graphics, and mathematics, a form of expression and computation. Rules of style for those forms apply to spreadsheets. For example, as with writing, a spreadsheet is easier to read if its text has proper spelling and grammar. As with graphics, a spreadsheet is easier to read if colour is used carefully. And as with mathematics, a spreadsheet should be easier to understand if the formulas are reasonably simplified.

With the gracious permission of Lindo Systems, Inc. (http://www.lindo.com), this paper uses an example spreadsheet called Assign.xls, distributed with What’sBest! (Lindo Systems, 1996), a commercial spreadsheet solver. The file can be downloaded with the student version of the company’s Solver Suite from http://www.lindo.com. Assign.xls is a simple employee scheduling problem.

In the following, a numeric cell is a formula or a constant referenced by formulas in other cells. A label, even if it is a numeric constant, is not a numeric cell. Also, we will avoid the word "user." The analogy to writing provides clearer terminology—writer and reader. You will see Excel formulas of the form WB(*a*, "operator", *b*), where "operator" is "=", "<=", or ">=". This is how What'sBest! defines a constraint, where *a* is the left-hand side and *b* is the right-hand side.

This paper is organised as follows:

Organize the spreadsheet for readability.
Be concise.
Format for description, not decoration.
Expose rather than hide information.

2. Organize the Spreadsheet for Readability

2.1 Make a Spreadsheet Read Left to Right and Top to Bottom

Gopen and Swan (1990) used linguistics and cognitive psychology to study scientific writing. They wrote, "Since we read from left to right, we prefer the context on the left, where it can more effectively familiarize the reader. We prefer the new, important information on the right, since its job is to intrigue the reader. Information is interpreted more easily and more uniformly if it is placed where most readers expect to find it." Gopen and Swan are echoed by Cohen (1997) and Microsoft (1995, p. 384).

In a spreadsheet, old information is the input data, since the writer expects the reader to know it. The new information is the formula, the derived information that the reader seeks. We expect to see the data first, and when we have digested that, we expect the output formula, to the right or below nearby. Intermediate formulas logically are data for later formulas, so the rule applies recursively to all numeric cells.

Exceptions depend on reader expectations. An accounting balance sheet typically has years in columns. A year's profit in one column at the bottom may flow to the next year in the next column at the top. The reader of accounting may reasonably expect one year's bottom line to flow into next year.

Archer (1989) and Davis (1996) observed that cell relationships can be represented as a directed graph, which Archer called a cell relationship diagram. Excel's auditing toolbar has buttons to insert temporary arrows that point to the dependents or precedents of a cell. We will call these graphical segments—Archer's cell relationship diagram—the arcs of precedence. A spreadsheet reads from left to right and top to bottom if every numeric cell's arcs of precedence start above and to the left of the cell.

Clicking several times on the Trace Precedents button displays the complete precedence tree of the cell. This tree can be quite illuminating. If a precedence tree is tangled like a bowl of spaghetti, the spreadsheet legitimately can be called a spaghetti spreadsheet! If a spreadsheet's precedence tree includes blank cells, the spreadsheet is perverse, since it depends on information that is missing.

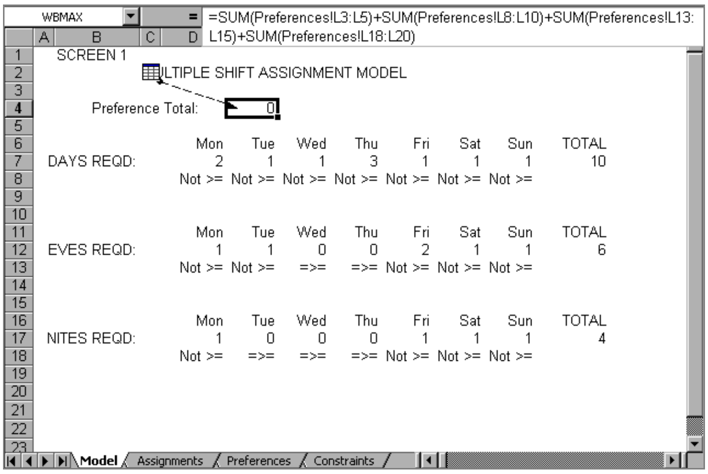
Figure 1 shows the Assign.xls Model sheet. The first numeric cell is the Preference Total cell, but no other information about preferences appears on the Model sheet. After hunting, we find the Preferences sheet listed at the bottom. Assign.xls has the objective function at the top and front, the way one might write a Lindo model (Lindo Systems, 1996). Preference Total is the objective function because E4 has range name WBMAX. The precedence arc is no help; all precedents of this cell are on another sheet.

2.2 Have Short Arcs of Precedence

Old information should be followed immediately by the related new information (Gopen and Swan, 1990). Higham (1993, p. 15), writing about mathematics, stated, "Try to minimize the distance between a definition and its place of first use." Conway and Ragsdale (1997) wrote, "Things which are logically related... should be arranged in close physical proximity and in the same columnar or row orientation."

If a cell is close to its dependents, the reader will more easily see the relationship between them. The spreadsheet will be naturally organised by blocks of meaning, blocks that reflect the business rather than the mechanical requirements of the spreadsheet. By contrast, arranging by input-calculation-output separates related cells by a large visual distance.

Figure 1. Assign.xls, Version 1. The Preference Total precedence arc points meaninglessly off screen. The Preference Total cell is too far from its precedents.



3. Be Concise

Regarding text, Strunk and White (1979, p. 23) wrote about text, "Vigorous writing is concise." Higham (1993, p. 25) wrote about mathematics, "Do not use mathematical symbols unless they serve a purpose." Tufte (1983, pp. 51, 100) wrote about graphics, “Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space…” Daellenbach (1994) wrote about math modelling, “A good model is a model that is as parsimonious as possible in terms of the variables/aspects included.”

None of the spreadsheet literature was concerned with this; that a spreadsheet should be concise is entirely novel and controversial. But the literature shows clearly that about 2% of spreadsheet cells are wrong (Panko and Sprague, 1998). Reducing the number of cells therefore may increase the probability of a correct spreadsheet.

3.1 Be Concise with Sheets

If it will fit on one sheet, consider putting it on one sheet. This is controversial—other writers encourage the use of multiple sheets, with little justification (e.g., Mather, 1999).

Multiple sheets breed unnecessary “spurious” cells and labels, as the writer copies information from other sheets. A spurious cell has exactly one precedent; it simply points to another cell, such as the formula “B25=A10” (not in Assign.xls). Mathematically, this is the same as $a = b$. A writer wants to remind the reader of data from another section; later formulas refer to the spurious reference rather than the original data. Such copying can duplicate the entire “input sheet” on every other sheet! Now the reader feels the temptation to check back and forth, to verify the formula visually. Various bad habits motivate the use of spurious references, but mainly the problem is from long arcs of precedence or multiple sheets.

Multiple sheets dislocate related blocks; the reader must remember a context from one screen to the next. When moving from one screen to the next, the reader worries, “I hope it’s laid out the same as the previous one I just spent all this time learning.”

Multiple sheets are hard to navigate. Many people do not know the key sequence for changing sheets (Ctrl-Pg Up, Ctrl-Pg Dn) and therefore must use the mouse, but most can use keys to move around a single sheet.

Multiple sheets make information harder to find. The search function in Excel does not scan different sheets by default, only the selected sheet. To select several sheets at once, the reader must click each sheet tab while pressing Ctrl. Most people will not know such arcana.

Multiple sheets render auditing tools useless, as we saw above. An arc of precedence to off-sheet cells displays an uninformative “off the sheet” icon, so the reader cannot visually see how cells are related.

Multiple sheets make it hard to find blanks. Writers frequently do not bother to delete blank sheets in the file (Excel defaults to 16). The reader must check the blank sheet to see if it contains data, a waste of time.

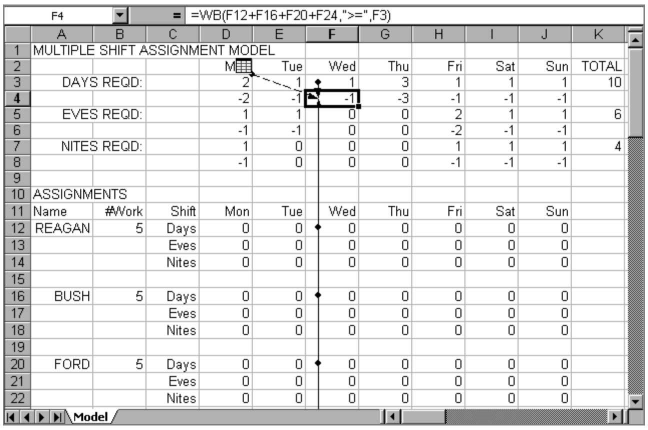
Spreadsheet writers tend to overlook these problems with multiple sheets for the sake of “modularity.” Of course, many times a modular approach may be called for. However, the writer can often reduce considerably the size of a large spreadsheet by moving everything to one sheet and then eliminating the spurious cells.

3.2 Use a Minimum of Blank Space and Only to Divide Blocks Visually

Empty cells are not equivalent to empty space in graphics. Blank space in a picture does not hide anything and is not accidentally covered by nearby information. Its effect is purely visual. By contrast, a spreadsheet cell is a discrete object. It can be formatted as hidden. It can be blank, but perversely another cell can depend upon it. A blank cell may appear to contain data that is really in a different cell.

Though empty cells are not equivalent to blank space in a graphic, empty cells can be used in the same way if the writer is careful. The writer should separate blocks with blanks as one separates sentences with a period or paragraphs with an indent. But arrange the data types so there is a reasonable minimum of punctuation. Small blocks can be adjacent without blank space between.

Figure 2. Assign.xls, Version 2, on one sheet



F4 = =WB(F12+F16+F20+F24,">=",F3)

	A	B	C	D	E	F	G	H	I	J	K
1	MULTIPLE SHIFT ASSIGNMENT MODEL										
2				M	Tue	Wed	Thu	Fri	Sat	Sun	TOTAL
3	DAYS REQD:			2	1	1	3	1	1	1	10
4				-2	-1	-1	-3	-1	-1	-1	
5	EVES REQD:			1	1	0	0	2	1	1	6
6				-1	-1	0	0	-2	-1	-1	
7	NITES REQD:			1	0	0	0	1	1	1	4
8				-1	0	0	0	-1	-1	-1	
9											
10	ASSIGNMENTS										
11	Name	#Work	Shift	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
12	REAGAN	5	Days	0	0	0	0	0	0	0	
13			Eves	0	0	0	0	0	0	0	
14			Nites	0	0	0	0	0	0	0	
15											
16	BUSH	5	Days	0	0	0	0	0	0	0	
17			Eves	0	0	0	0	0	0	0	
18			Nites	0	0	0	0	0	0	0	
19											
20	FORD	5	Days	0	0	0	0	0	0	0	
21			Eves	0	0	0	0	0	0	0	
22			Nites	0	0	0	0	0	0	0	

Model

Cells in rows 4, 6, and 8 refer to cells below, against the direction we read. B12, B16, and B20 also refer to cells below and are unrelated to cells nearby. The spreadsheet should be arranged so each cell depends only on cells above.

3.3 Keep Information in One Cell Logically in One Cell Visually

Make blank cells look blank. Make active cells look active. Try to avoid labels that overlap neighbouring cells. Leave on the default grid so cells are displayed.

In Figure 1, the Preference Total label could be in B4, C4, or D4. In fact, both C4 and D4 are blank. The label is in cell B4 with leading spaces. The leading spaces are unnecessary and should be deleted. The label should be put in cell D4, adjacent to the cell it labels.

For Figure 2, cut and paste was used to put Assign.xls on one sheet. Preference Total was moved to the bottom, so its precedents would be above it. Then Excel's default grid lines were turned on, and blank rows and columns were deleted. The spreadsheet now fits in three screens, even at low 640 by 480 resolution.

3.4 Be Concise with Blocks

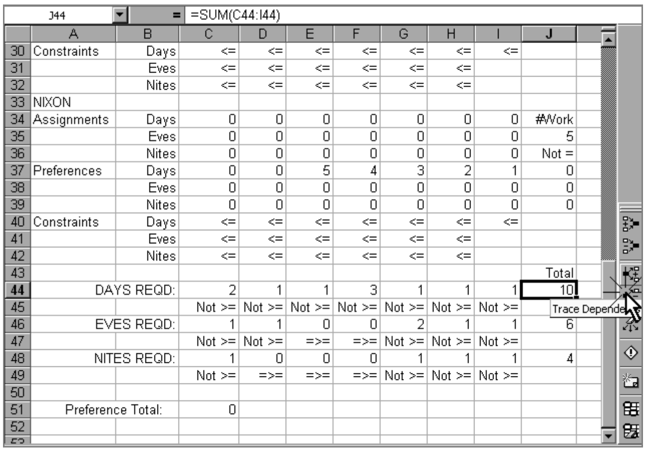
Align data types consistently. If "names" are listed column-wise in one place, then list them column-wise everywhere. This gets the reader oriented to viewing the spreadsheet in a consistent way. Align the primary data type in rows. Excel has 65,536 rows versus 256 columns, so if the primary data type is downwards, the writer is less likely to run out of room. More importantly, we read left to right, so the reader expects to see the table structure across the top, in the column labels. The repetitive information should be below, so the reader views it by paging down.

When a spreadsheet requires more than one table, stack unrelated blocks vertically (like a web page) or horizontally (like ticker tape), but not both (a bulletin board). Information one screen down and one screen right is hard to find, unless it is part of a table that starts on the far left. A bulletin board layout requires more keystrokes to navigate.

Conway and Ragsdale (1997) wrote, "A design that results in formulas that can be copied is probably better than one that does not." Four signs for finding such a design are time, label repetition, concreteness, and formula transposition. All numeric cells varying with the same time periods should be in a single table. Label repetition (e.g., the days of the week labels in Figure 1) suggests that the writer could consolidate blocks. A concrete data type such as "ex-president" is probably a better structure than an abstract data type, such as "data" or "constraints". Transposing with formulas is a sign that data types are misaligned.

Cells J44, J46, and J48 are dangling since they have no dependents. These dangling cells can be deleted.

Figure 3. Assign.xls, Version 4, bottom part, with the auditing toolbar



J44 = =SUM(C44:I44)

	A	B	C	D	E	F	G	H	I	J
30	Constraints	Days	<=	<=	<=	<=	<=	<=	<=	
31		Eves	<=	<=	<=	<=	<=	<=		
32		Nites	<=	<=	<=	<=	<=	<=		
33	NIXON									
34	Assignments	Days	0	0	0	0	0	0	0	#Work
35		Eves	0	0	0	0	0	0	0	5
36		Nites	0	0	0	0	0	0	0	Not =
37	Preferences	Days	0	0	5	4	3	2	1	0
38		Eves	0	0	0	0	0	0	0	0
39		Nites	0	0	0	0	0	0	0	0
40	Constraints	Days	<=	<=	<=	<=	<=	<=	<=	
41		Eves	<=	<=	<=	<=	<=	<=		
42		Nites	<=	<=	<=	<=	<=	<=		
43										Total
44	DAYS REQD:		2	1	1	3	1	1	1	10
45			Not >=	Not >=	Not >=	Not >=	Not >=	Not >=	Not >=	
46	EVES REQD:		1	1	0	0	2	1	1	6
47			Not >=	Not >=	=>=	=>=	Not >=	Not >=	Not >=	
48	NITES REQD:		1	0	0	0	1	1	1	4
49			Not >=	=>=	=>=	=>=	Not >=	Not >=	Not >=	
50										
51	Preference Total:		0							
52										

Assign.xls was originally structured as a linear program with decision variables, constraints, objective coefficients, etc. A client is unlikely to want to use it that way. The client instead will probably think about ex-presidents. For example, how can we add a new ex-president? Structured by decision variables, preferences, and constraints, the client must copy three sections separately. We will rearrange Assign.xls so the primary data type is an ex-president, and move the "#Work" cell closer to where it is used. If we want to add another ex-president, there is only one block to copy. The amount of visual space used by Assign.xls (as in Figure 3) is now two and a half screens, better than the original four sheets.

4. Be Concise with Cells

All cells in a spreadsheet, sometimes even blanks, require the reader's interpretation. Unnecessary cells take up the reader's precious time, time that the writer wants the reader to spend on understanding the business logic. Unnecessary cells can contain mistakes, add visual and logical clutter, add to the bulk of the file, and confuse the reader.

4.1 Eliminate Spurious References (Described Above)

Fortunately, Assign.xls has none.

4.2 Erase Dangling Cells

A dangling cell is a numeric cell without dependents, a calculation not used anywhere else. A cell not on the precedence tree for the bottom line is dangling. Again, following Archer's (1989) cell relationship diagram, dangling cells will be leaves on the graph. Usually, a spreadsheet contains a "bottom line," such as an objective function or balance sheet total profit. Except for this, dangling cells are usually unnecessary.

Just as a spurious reference is analogous to redundant text, a dangling cell is analogous to irrelevant text, an unused variable, or an unneeded decoration. There are four main types of dangling cells: the unused input, the validation formula, the useless intermediate calculation, and the interpreted output. Unused input is a relic from an earlier version or an error in the model. We discuss relics later.

Authors cited earlier recommended formulas to check the validity of input and output. These extra formulas are dangling by definition. Temporary validation formulas are useful for debugging, but reduce readability, and eventually should be erased.

Writers insert useless intermediate calculations, thinking the reader "will want to see a subtotal, just for their information." But the writer should instead drive home the main point rather than display trivial side matters. Let the readers put these in as they wish.

Interpreted output is like a comedian repeating the punch line, hoping for an extra laugh. For example, consider the formulas (not in Assign.xls):

```
D49: SUM(D3:D48)
D50: if(D49>0, "Surplus of " +TEXT(D49,0), "0")
D51: if(D49<0, "Funding gap of " +TEXT(-D49,0), "0")
```

Cells D50 and D51 merely interpret D49. Cell D49 should be labelled like "Surplus (gap)", and the two spurious formulas D50 and D51 should be erased. Avoid labels containing formulas. While they seem clever to the writer, the reader has a harder time distinguishing the numeric model from the documentation.

Does Assign.xls have dangling cells? In the NIXON block of Figure 3, cell range C40:H42 and cells I40, J35, and J36 do not appear on the precedence tree of the objective function Total Preferences. But they are required for the constraints, and are referenced by the What'sBest! solver. This is a compelling reason to leave them in. But the "Total" cells J44, J46, and J48 are danglers of the useless-intermediate-calculation variety, so they will be erased.

4.3 Simplify Formulas

For example, the formula (not from Assign.xls) C6*(A4) + A6*C6 + ((C6*A5)) could be simplified to C6*(A4+A5+A6), or C6*SUM(A4:A6), which changes automatically if a row were inserted. Use the fewest characters necessary to write the formula correctly. Table 1 summarises issues of formula readability.

Table 1. Factors in Cell Readability

Less readable	More readable
Irregular terms reference various ranges. References appear more than once.	Similar terms repeat and reference similar ranges.
References are ordered randomly in the formula.	References read left to right in row and column order.
Well-known formulas are made cryptic by separation into multiple cells.	Well-known formulas (e.g., the quadratic formula) are in a single cell.
Division appears randomly.	Division generally appears at the end.
Formulas contain constants or reference blanks.	Formulas reference constants.
Formulas have unnecessary parentheses and spaces; the formula can be simplified.	Formulas have the fewest characters needed.

4.4 Nest and Erase Formulas Where Appropriate

Sometimes making a few cells slightly more complicated allows many cells to be erased. Nesting is especially appropriate when a formula in cell x has only one dependent, cell y. The formula in x can be copied and substituted for the address of cell x in cell y. This process should stop when the formula in the dependent cell y begins to lose readability.

Other authors (e.g., Freeman, 1996, and Mather, 1999) say the opposite: complicated formulas should be separated into multiple cells. However, nesting of formulas follows from mathematical writing, which prescribes substituting out unnecessary variables. Virtually every other form of expression favours brevity. The writer should weigh the number of cells against the readability of individual formulas, but the error thus far has been on too many cells.

Many cells can be eliminated with the sumproduct() formula. In Assign.xls, Figure 3, cells J37:J39 multiply preferences by assignments and then are summed in

Preference Total. Cells J37, J38, and J39 could be replaced with one sumproduct(), and then nested directly into Preference Total. We can do the same substitution with the other blocks. The Preference Total becomes

C51: SUMPRODUCT(C4:I6,C7:I9) + SUMPRODUCT(C14:I16,C17:I19) + SUMPRODUCT(C24:I26,C27:I29) + SUMPRODUCT(C34:I36,C37:I39)

This allows us to erase twelve cells, 4% of the numeric cells. All precedents of this cell are now constants and may be viewed in one click. Hence, it will be easier to debug and edit. While long, this formula is readable because it is regular.

4.5 Eliminate Relics from Earlier Versions

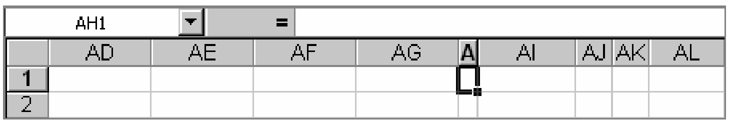
A relic is anything no longer needed but appears because the writer has not cleaned it up. Relics confuse the reader.

A simple way to look for spreadsheet relics is to press the End key, then the Home key. This moves the cursor to the lowest and furthest right cell for which Excel has allocated memory. If this last cell is not the bottom right cell of the intended spreadsheet, then there may be relics. Most of the time, extra rows and columns contain old formats.

A fast way to see if apparently blank rows or columns affect the spreadsheet is to delete them. If #REF! errors appear, undo the change and find the problem; formulas probably refer to blank cells.

Assign.xls contains relics of earlier versions. In the Model sheet, the data seem to be in columns A through J. But width was adjusted of columns as far right as AU. In fact, in the original version, pressing End and Home moves the cursor to cell IT22, far away from the apparent last cell in L18. Fortunately, these relics are just old formats, as we see in Figure 4.

Figure 4. Assign.xls Contains Relics, unneeded formats from earlier versions



5. Format for Description, Not Decoration

A format is descriptive if it displays information and if the reader already knows the format. A format is decorative if it displays no information. Writers format cells for various reasons: to get attention, to try to look professional, from perceived necessity, to encode data types, and to make a cell self-descriptive. These overlap somewhat and are variously decorative or descriptive. The worst are cryptic; the best are crucial. In the following, we applied to spreadsheets Tufte's (1983, p. 183) suggestions about graphics.

Use formats to get attention or to try to look professional very sparingly. For the writer, default formats take no time to apply. For the reader, the default format is best in most cases. Use only one font size. Large fonts take up screen space, take time to produce, and require changes in row heights which affect the entire spreadsheet width. Avoid using many colours; readers bring their own complex interpretations and perceptions of colours and may be colour blind. Avoid multiple styles in one cell—e.g., bold and italic—and use these rarely. Underscored spreadsheet text is ugly. To get attention on a few specific cells, use a logical layout and concise labels (Tufte, 1983, p. 183).

Use formats of perceived necessity very sparingly. The most common format of necessity is the column width. Except for in column A, which should contain labels, try to give all columns about the same width. Many different column widths can be disconcerting; the reader may try to find meaning in a cell width when there is none. Uniform columns are more aesthetically pleasing. Avoid varying row heights. A thick row with word wrap will tend to force the reader to read primarily downwards. For the same reason, do not rotate fonts (Tufte, 1983, p. 183). Do not shrink fonts either – the reader's display may be small.

Format constant data differently from formulas. The most crucial structural distinction is the one between constants and formulas (Conway and Ragsdale, 1997). Constants can be interpreted faster than formulas, so the reader will see obvious constants with relief. Without a compelling reason, do not try to encode data types beyond the distinction between constants and formulas. Additional encoding will not naturally convey information and can easily become cryptic (Tufte, 1983, p. 183). Edwards and Finlay (1997) also distinguished between constant data and formulas by protecting formulas to prevent accidentally overwriting a formula.

Figure 5. Assign.xls, Final Version. Abbreviations were eliminated, formulas are grey

J9 =WB(SUM(C7:I9),"=",J8)

	A	B	C	D	E	F	G	H	I	J
1	Multiple shift assignment model							Formulas are grey.		
2										
3	Reagan		Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	
4	Preferences	Days	1	0	3	2	0	0	0	
5		Evenings	0	0	0	0	4	5	0	
6		Nights	0	0	0	0	0	0	0	
7	Assignments	Days	0	0	0	1	0	0	0	Shifts/week
8		Evenings	0	0	0	0	1	1	1	5
9		Nights	1	0	0	0	0	0	0	=
10	Constraints	Days	=<=	<=	<=	=<=	=<=	=<=	=<=	
11		Evenings	=<=	<=	=<=	<=	=<=	=<=		
12		Nights	=<=	<=	=<=	=<=	=<=	=<=		
13	Bush									
14	Preferences	Days	3	2	0	0	0	0	0	
15		Evenings	0	0	1	0	0	0	0	
16		Nights	0	0	0	0	5	4	0	
17	Assignments	Days	1	1	0	0	0	0	0	Shifts/week
18		Evenings	0	0	0	0	0	0	0	5
19		Nights	0	0	0	0	1	1	1	=
20	Constraints	Days	=<=	=<=	<=	<=	=<=	=<=	=<=	
21		Evenings	=<=	<=	<=	<=	=<=	=<=		
22		Nights	=<=	<=	<=	<=	=<=	=<=		
23	Ford									
24	Preferences	Days	5	0	3	4	0	0	0	
25		Evenings	0	0	0	0	2	1	0	

Following the principle of minimal formatting, either constants or formulas may be formatted—whichever results in the least formatting—with a light grey background colour and a thin grey border. A grey background is easy to notice, photocopies better than a coloured font and is not distracting. In any case, document the difference on the spreadsheet with a formatted label—e.g., "Formulas are grey."—as in Figure 5.

Use formats of self-description liberally. Format numbers according to their meaning using widely understood conventions to make information self-descriptive. Right justify numeric cells and the labels above them. All cells of a given data type should display the same number of decimal places—the reader expects a number's width to indicate its magnitude. If the number is a percent, format it as a percent, so it displays "%". If the number is money, format it to display a monetary symbol, such as "$". Show separators—e.g., "1,000,000".

6. Expose Rather Than Hide Information

Put labels in the spreadsheet, and make sure most of them are on the left. On mathematical writing, Higham (1993, p. 25) recommended, "Avoid starting a sentence with a mathematical expression, particularly if the previous sentence ended with one, otherwise the reader may have difficulty parsing the sentence."

Spell out labels and use the spell checker. Abbreviations inhibit comprehension (Tufte, 1983, p. 183). Even supposedly "commonly understood" symbols (such as Q for the order quantity, A for order cost) should be written out in a spreadsheet (Order quantity Q, Order cost A).

Use proper case. Text is harder to read if written in capitals (Tufte, 1983, p. 183). It is harder to find the beginning and end of a sentence, and it is not what readers expect. Capitals are wider than lower case, so less text can fit in the width of a cell. The last version of Assign.xls appears in Figure 5 with proper case.

Of course, there is the old adage, "Do not put constants in a formula," since data in a formula hides information rather than exposes it, as in Table 2. But there is a more powerful rule: try to make every formula reference only constants.

Table 2. Comparison of Friendly and Unfriendly Cell Formats

unfriendly	friendly
B5: 100000	B5: 7%
B6: 20	B6: 20
B7: PMT(0.07, B6, B5)	B7: $100,000
	B8: PMT(B5, B6, B7)

Table 2 summarises unfriendly and friendly—i.e., hidden versus exposed information—cell formats. On the left, 0.07 should be in its own cell; formula references to B5, B6, and B7 should be in order, and values should be formatted.

A spreadsheet with hidden cells is perverse because hidden cells are inaccessible dependents. Hiding cells or preventing changes in a spreadsheet is irritating and tends to reduce the reader's confidence in the spreadsheet. If a reader takes the time to audit a spreadsheet with hidden cells, the model cannot be proved correct because the reader cannot see the formulas. The writer assumes the reader cannot improve the work, and preventing derivations from it diminishes its utility.

The logic is literally hidden. Developers will argue that cells should be hidden in corporate spreadsheets, which have data entered by various clerks and managers. While this may be true, it should be the exception rather than the rule.

Some writers feel the need to "idiot proof" a spreadsheet. This usually involves lots of formatting, heavy lines around the inputs, many colours, an input sheet, a summary sheet, password protection, etc. Instead, have the fewest cells necessary to produce the result, flow the logic from top to bottom and left to right, and put related cells close together. And put it all on one sheet. "Keep it simple" could be restated as "Keep it small."

7. Conclusion

While we have used analogies to writing text, mathematics, and graphics, the analogy to writing text is perhaps the best. Creating a spreadsheet is more like writing text than it is like coding in Fortran. We should tell students not to embed numeric constants in formulas, but we might explain it by saying "because you should not hide a key definition in a footnote or appendix." Hopefully, this paper has helped to debunk the idea that a "spreadsheet as computer program" structure will improve spreadsheet readability and reduce spreadsheet error. In its place, we developed a new style for writing spreadsheets based on writing text, graphics, and math models.

References

Archer, N. P., (1989), "Electronic Spreadsheet Structures," *Computers & Operations Research*, 16(5), 493-496.

Bissell, J. L., (1986), "Spreadsheet Planning and Design," *Journal of Accountancy*, 161(5), 110-120.

Bromley, R. G., (1985), "Template Design and Review: How to Prevent Spreadsheet Disasters," *Journal of Accountancy*, 160(6), 134-142.

Cohen, M., (1997), "Auditing Spreadsheets," *Accountancy Ireland*, 29(1), 12-13.

Conway, D. G. and C. T. Ragsdale, (1997), "Modelling Optimization Problems in the Unstructured World of Spreadsheets," *OMEGA*, 25(3), 313-322.

Cragg, P. B. and M. King, (1993), "Spreadsheet Modelling Abuse: An Opportunity for OR?" *Journal of the Operational Research Society*, 44(8), 743-752.

Crain, J. L. and W. C. Fleenor, (1989), "Standardizing Spreadsheet Designs," *CPA Journal*, 59(10), 81-84.

Daellenbach, H. G., (1994), *Systems and Decision Making*, Chichester, England: John Wiley Ltd., p. 132.

Davis, S. J., (1996), "Tools for Spreadsheet Auditing," *International Journal of Human Computer Studies*, 45, 429-42.

Edge, W. R. and E. J. G. Wilson, (1990), "Avoiding the Hazards of Microcomputer Spreadsheets," *Internal Auditor*, 47(2), 35-39.

Edwards, J. S. and P. N. Finlay, (1997), *Decision Making with Computers: The Spreadsheet and Beyond*, London: Pitman Publishing.

Edwards, J. S., P. N. Finlay, and J. M. Wilson, (2000), "The Role of OR Specialists in 'Do It Yourself' Spreadsheet Development," *European Journal of Operational Research*, 127, 14-27.

Freeman, D., (1996), "How to Make Spreadsheets Error-Proof," *Journal of Accountancy*, 181(5), 75-77.

Gopen, G. D. and J. A. Swan, (1990), "The Science of Scientific Writing," *American Scientist*, 78, 550-558.

Higham, N. J., (1993), *Handbook of Writing for the Mathematical Sciences*, SIAM.

Kee, R., (1988), "Programming Standards for Spreadsheet Software," *CMA Magazine*, 62(3), 55-60.

Kee, R. C. and J. O. Mason Jr., (1988), "Preventing Errors in Spreadsheets," *Internal Auditor*, 45(1), 42-47.

Lindo Systems, Inc., (1996), *Solver Suite: LINDO, LINGO, and What'sBest*!, Chicago, Illinois: http://www.lindo.com.

Mather, D., (1999), "A Framework for Building Spreadsheet Based Decision Models," *Journal of the Operational Research Society*, 50, 70-74.

Microsoft Corporation, (1995), *Windows Interface Guidelines for Software Design*, Redmond, Washington: Microsoft Press.

Panko, R. R. and R. H. Sprague Jr., (1998), "Hitting the Wall: Errors in Developing and Code Inspecting a 'Simple' Spreadsheet Model," *Decision Support Systems*, 22(4), 337-353.

Stone, D. N. and R. L. Black, (1989), "Using Microcomputers: Building Structured Spreadsheets," *Journal of Accountancy*, 168(4), 131-142.

Strunk, W. and E. B. White, (1979), *The Elements of Style*, 3rd ed., New York: MacMillan Publishing Company, Inc.

Tufte, E., (1983), *The Visual Display of Quantitative Information*, Cheshire, Connecticut: Graphics Press.

Winston, W., (1993), *Operations Research*, *Applications and Algorithms*, Belmont, California: Duxbury Press.

Winston, W., (1996), "The Teachers' Forum: Management Science with Spreadsheets for MBAs at Indiana University," *Interfaces*, 26(2), 105-111.