FrameMaker 4

Graeme Forbes

FrameMaker 4 is the latest version of Frame Technology's multiplatform "structured document" publishing program. It runs on the Apple Macintosh (a native version for PowerPC Macs was due in mid-April), in Microsoft's Windows environment for Intel PCs, and on various UNIX systems. The program runs identically on all platforms; indeed, because Microsoft's Windows interface is so similar to the Mac's, Frame's documentation is the same for both versions of the program. UNIX users do have a little more flexibility in customization, though. For example, at the system level they can change the grey percentage of the default fills in the program's graphics module, while to get the same effect, Mac users have to edit the code of their individual files after saving them in a special format. Nevertheless, this review will focus on *FrameMaker 4* for the Mac, specifically the maintenance release 4.0.2.

FrameMaker 4 is a desktop publishing (dtp) program. The desktop publishing market was created by the Macintosh and especially by the introduction of the LaserWriter printer and the release of Aldus's *PageMaker* in 1985. A dtp program differs from a word processor in that it allows you to format text in layout arrangements of your choice, whereas, at its simplest, a word processor merely allows you to type text across the page, line after

line, and format it at the level of words and paragraphs. However, with each upgrade of a modern high-end word processor, such as (on the Mac) *Nisus*, *Word* or *WordPerfect*, more and more dtp capacities get added, so that the differences between the two types of program are shrinking in number. They are still significant, however, and the best-known dtp programs, *PageMaker* and *QuarkXPress*, concentrate on layout facilities and expect that most of the text to be incorporated will have been imported from a word processor. Another difference between word processors and dtp programs is that the high-end versions of the latter list at around twice the price of the high-end versions of the former. Fortunately, Frame Technology offers good academic discounts that make *FrameMaker 4* much more price-competitive.

FrameMaker does not compete with *PageMaker* or *QuarkXPress*; indeed, with the demise of *Interleaf* on the Mac, *FrameMaker* has no real Mac competition among dtp programs. *PageMaker* and *QuarkXPress* are best suited for short design-intensive documents, while *FrameMaker*'s forte is in the creation of long multi-chapter documents which require consistent and complex kinds of formatting (*PageMaker* has some, but not enough, of the necessary tools). The most frequently cited example of what you would want to use *FrameMaker* for is technical documentation, for example, hardware and software manuals of the kind the reader is no doubt familiar with. And though it is quite possible to create the text for such a book in a word processor and then import it into *FrameMaker*, *FrameMaker* comes with its own array of word processing features, and users who take to it may find themselves fairly quickly abandoning their word processors. *FrameMaker*'s capacities make it an excellent choice for dictionaries, handbooks, college textbooks, edited editions of literary works, and for research monographs

in some of the more technical humanities, such as linguistics and some areas of philosophy, as well, of course, for documents in the sciences.

Document Structure

A printed page from a *FrameMaker* file is a composite of elements from electronic pages of as many as three types: in addition to a body page, which holds the main text, a "master" page and perhaps also a "reference" page are involved. A master page contains elements that repeat from page to page in the printed output, for instance, the document title, the author name, and the page number. Master pages also contain text frames that constitute a template column layout for body text, and may contain a "watermark," typically a word or phrase such as "Draft," which will print in light grey behind the body text. A *FrameMaker* document can contain up to one hundred different master pages, which should be more than enough.[1]

A normal arrangement for a document with a single column of body text on each page is to have one or two main master pages (two if one wishes to differentiate left and right printed pages, say, by the positioning of header text); a title master page, on which the body text begins lower on the page and there is perhaps some graphic at the top of the page; a variant title master page with a watermark on it; and special master pages for an index, a table of contents, or a bibliography. As you enter text *FrameMaker* automatically adds new body pages which are assigned a default master page. Unfortunately, you cannot change this default. But at any point you

1. I used *FrameMaker* to prepare camera-ready proofs of my textbook *Modern Logic* for Oxford University Press. Though the book is 397 pages long, I used only twelve different master pages, including three special ones for the index.

can assign a different master page to an existing body page, or a particular master page to a range of existing body pages, or to the odd pages or even pages within the range. And you are never committed to the look of a particular master page. If you make a change to a master page, all body pages assigned that master page can be updated at once. Alternatively, you can change, or "override," certain master-page elements from within a particular body page and then decide whether to update all other body pages that use the corresponding master page, to create a new master page using the altered body page, or to retain the current master page with its overrides on the body page. So there is considerable flexibility in this method.

The text columns that appear on master pages are assigned to "flows," which control what happens when a particular column on a body page has been filled with text. If your pages have two text columns, you may want text on the first column of a given page to flow into the first column on the next page, for example, if the first column contains text in one language and the second its translation in another. This type of flow is achieved by assigning two different "tags" to the text columns on the master page; then, if "Autoconnect" is on (which causes *FrameMaker* to add new pages as required), text in a column with the tag "B" will flow onto the next page into the column tagged "B." Furthermore, there is a battery of other operations for columns: synchronizing, feathering, splitting, joining, and so on.

A reference page contains graphical elements, either created in *FrameMaker* or imported from a graphics program, that you wish to repeat in your document in some systematic way. For example, if you wish to separate the first footnote on a page from the main text by a rule of a certain width and thickness, you would create it on the reference page (or customize a rule

already drawn there), name it, and instruct the program to insert it automatically at appropriate places. A document can have up to a hundred reference pages, though a document that required more than one or two would be graphically very elaborate.

Text Editing

FrameMaker is designed to make a separate word processor redundant, and certainly some of its best features are in the realm of text editing. However, its worst features are also in the realm of text editing , and it is as well to get the bad news out of the way. *FrameMaker*'s single worst feature is its note handling: (1) it does not do automatic endnotes (although there is a clumsy approximation, which I describe in note 4); and (2) its automatic footnotes are incorrectly implemented—specifically, if a footnote referred to on page *n* is too large to fit on that page underneath the text line where the reference occurs, the *entire* note is moved to the next page (with the obvious potential knock-on effect) instead of being broken across pages *n* and *n*+1. Many academic readers of this review, having taken in the previous sentence, are likely to lose all interest in *FrameMaker* and turn to another article in this issue of *Text Technology*. It is remarkable that a company offering academic discounts, and so presumably seeking a good share of the academic market, should continue to sabotage itself in that arena by letting *FrameMaker* reach version 4 without bothering to fix a facility which is so important to most of its potential academic customers, meanwhile expending programmer time on peripheral and duplicative features (e.g. the QuickAccess [sic] bar). On the Mac, *FrameMaker* is potentially the most attractive document creation program for an academic, and its note handling is the

biggest obstacle to its wider acceptance.[2]

The other jarring deficiency in *FrameMaker*'s text editing capabilities, especially for a user coming to the program from *Nisus*, is its weak Undo facility. There are far too many operations that cannot be undone, and even for those that can be, there is only one level of Undo. Nowadays a good text editor ought to be able to undo every change made since the last opening of the document, even through file saves, but on the Mac only *Nisus* offers this facility, though some graphics programs have at least an adequate number of undo levels. Maximum undoability is possible only if every editing operation is undoable, but in *FrameMaker*, absurdly, if you perform even the simplest global replacement, the result is not undoable—a disaster if you inadvertently used an incorrect text element that also occurs correctly throughout the document (if you try to repair the damage with another global replacement you will affect the correct occurrences as well).

However, *FrameMaker* also has significant strengths in text editing. In using a word processor, one tends to format text in a more or less ad hoc way. You may decide to set off a lengthy quotation from the main text by inserting extra space above and below and/or by altering the line indents. But if you don't take another look at this paragraph when you come to the next lengthy quotation, you may apply different spacing and indentation. In *FrameMaker* , however, you would use the "paragraph designer" to assign a tag (say "longquote"), and a format to go with that tag, to the first quota-

2. The lack of endnotes in a program whose output will be used as page proofs is defensible, since endnotes are such a reader-hostile way of organizing notes. But commercial publishers who will impose their own format often ask that notes be separated from the main text, even if they will end up as footnotes in the final product. And sometimes one would like both a footnote series and an endnote series in the same document.

tion. The format is stored in a "format catalog" and when you encounter the next lengthy quotation, if you assign it the tag "longquote" from the catalog, it will immediately take on the same format as the previous quote. The advantages of this method (which I believe first appeared on the Mac in *Word* as "stylesheets") are obvious: documents achieve an aesthetically pleasing consistency, and more pragmatically, if you become dissatisfied with the appearance of your long quotes, you can change them all at once simply by resetting parameters for the "longquote" tag in the paragraph designer. And if you want to adjust only a particular long quote—say, because the amount of space above it is forcing its last line onto the next page—then you can make changes in the designer but apply them to the currently selected paragraph only, without updating all other paragraphs that have the same tag. Such changes are called "overrides" and if you later want to change a different aspect of the appearance of all long quotes, you have the option of preserving any overrides you have already made.

Since the paragraph designer is at the heart of *FrameMaker*'s text editing, I will describe it in some detail. Designing is done in a window that can display any one of six different sets of properties; a pop-up menu in the window allows you to choose which set to display, or you can cycle through them with Page Up and Page Down. The six sets are: (a) basic; (b) default font; (c) pagination; (d) numbering; (e) advanced; and (f) table cell.

(a) Basic. In this window, shown in Figure 1, you can set values for the left indent of the first line, the left indent of subsequent lines, and the right indent, all relative to the edges of the text column; for the line spacing, the amount of extra space above and below a paragraph; for the positions and kinds of tab stops (left, center, right, decimal) and their leaders, if any (typ-

ically dotted or dashed lines); and for the kind of alignment you want the text to have (left, right, centered, justified). If you wish paragraphs of the type you are designing always to be followed by a specific but different type of paragraph, you can set a "Next ¶ Tag" in this window as well (a paragraph tagged "Title" should perhaps always be followed by a paragraph tagged "Author," which would have different properties).[3]

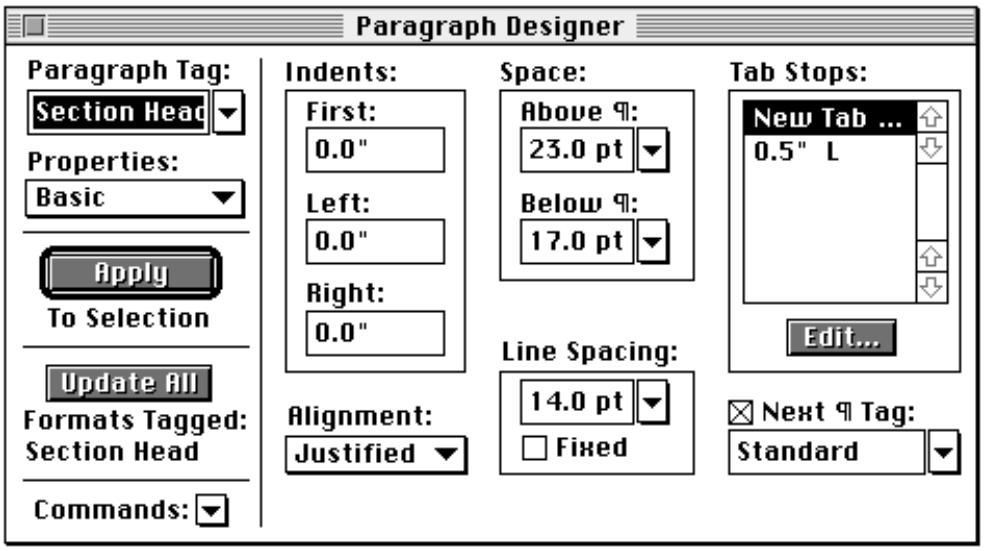

Figure 1: The Paragraph Designer window showing the Basic Properties view for a paragraph format called "Section Head." The drop-down Paragraph Tag menu lists all formats in the catalog. The drop-down Properties menu lists all six sets of properties. The Commands drop-down menu allows creation and deletion of new formats and various shortcuts for window settings.

(b) Default Font. In this window you set the default font family for the paragraph, along with point size, angle (regular, italic, oblique, cursive, etc.), weight (light, roman, demi, bold, black, ultra, etc.), variation (condensed, engraved, expert, and in some technical fonts, different ranges of math symbols), color, and spread. Spread determines paragraph letter spac-

3. *FrameMaker 4* can display a word processor-like formatting bar at the top of the screen, from which many settings can be made. But the designer windows offer more control, and the formatting bar takes up space that is better used to display another line of text.

ing and can be tightened or slackened according to font-metrics and point size. There are also check boxes for various special styles: underline, overline, outline and so on, and for activating automatic kerning. Of special note is the change bar style: when this is on, then if any change is made to text in the current paragraph, the entire paragraph will be flagged with bars placed in the margin.

The font properties are the ones that are most commonly overriden, especially by writers using foreign-language symbols or phonetic or math symbols. You can override default font properties in an ad hoc way by simply making changes from the menus, using Command-Shift-Space to reset the defaults. But if the kind of overrides you apply are fairly consistent, it is better to use the "character designer," a version of the paragraph designer Default Font window, in which you can create tagged character styles. Like paragraph tags, these character tags are stored in a catalog. The spread setting can be applied at the character level as well as the paragraph level and so allows for manual kerning. Typical entries in a character format window are "superscript roman," "superscript italic," "subscript roman" and "subscript italic." There is a generic "As is" setting which means, for example, that you would not have to create two different "subscript italic" character styles according to whether the default paragraph font is, say, serif or non-serif: by setting the font family for "subscript italic" to As is, you can ensure that the subscript will retain its typeface when the style is applied to existing text and will assume the typeface of the previous character when the style is applied before typing. If you have the screen space, you will want to keep the paragraph and character catalogs open, but if not, you can choose them from submenus in the format menu, or by *FrameMaker*'s own key-

board method, or most conveniently, by command key equivalents using a macro program such as *Tempo II* or *QuickKeys*.

(c) Pagination. This window is for properties that determine aspects of the positioning of a paragraph. You have the options of requiring that a paragraph begin at the top of a page, a column, a right page, or a left page; you can specify that it not be allowed to move to a different page from the previous or following paragraph; and you can set the allowable number of widow/orphan lines. *FrameMaker 4* adds some new options, of which "Run-in Head" is the most interesting. A run-in head is a paragraph ending with the standard end-of-paragraph symbol, yet the text of the next paragraph continues on the same line. This facilitates the composition of dictionaries or directories in which the headers are automatically updated when entries are added or deleted. You create these dynamic headers on the master page by inserting one or two special "variables" that point to a particular paragraph tag. Then, on each associated body page, *FrameMaker* replaces these variables with the text of the first or last paragraph currently on the page that has the specified tag. Dynamic headers like this were possible in *FrameMaker 3* but only at the cost of beginning the text of each entry on a new line so that the entry title would be a separate paragraph, which is necessary because the text of the entire paragraph is substituted for the variable. The run-in head style lets the text of the entry follow on the same line as the entry title, as in a typical dictionary.

(d) Numbering. In this window you can specify an "autonumber format" for a paragraph. For many academics it will be this feature, and the way it interacts with cross-references, that makes *FrameMaker*'s footnote/endnote inadequacies worth living with a while longer. A paragraph for-

mat can include a counter so that successive paragraphs in the document with that format are consecutively numbered, either at their beginnings or ends. This "autonumber" can be made up of Arabic numerals, Roman numerals, or letters (both cases), in any combination; it can contain tabbed text, and its font can be styled independently of the paragraph text. A number sequence can be restarted at any point, need not begin with "1" or "a," and can consist of various subseries. Thus one can easily build an autonumber format, with any number of levels, that produces numbering such as 1.1, 1.1.1, 1.1.2,…, 1.2.1, etc. Whenever you insert or delete a paragraph all the numbers are automatically adjusted. Best of all, by employing "series labels" one can have multiple autonumbered series in the same document, using the same or different styles of counter. So one might have paragraphs of body text autonumbered as above, examples autonumbered (a), (b), (c), etc., and figures autonumbered (1.1.i), (1.1.ii), etc. Anyone who writes frequently-revised papers with numbered series in them that have to be maintained manually will find *FrameMaker*'s autonumbering capacities outstanding.

(e) Advanced. This is the last paragraph format property set I will describe at this point. Here you can control aspects of hyphenation such as the maximum number of consecutive lines in the paragraph that can end in a hyphen, the shortest word that can be hyphenated, and the shortest prefix and suffix hyphenation can produce. You can also assign one of the graphic elements on a reference page to appear above or below the paragraph. And you can set a range for word spacing (as opposed to spread, which is letter spacing), allowing *FrameMaker* leeway in either direction from the word spacing specified by the metrics of the default font. Sometimes a word can't

be fitted on a line and can't be hyphenated, resulting in gaps between words larger than the spacing settings officially permit, but if "Allow Automatic Letter Spacing" is on, *FrameMaker* will add some space between letters to compensate. However, mindful of Goudy's dictum that anyone who would letterspace lowercase would steal sheep, typographic purists will prefer to keep the setting switched off and rewrite text if the problem arises.

Some of the features I have described pay more attention to the aesthetic qualities of printed output than the typical user of a word processor may care about. Such a writer can leave these settings at their defaults—they are there to be adjusted should he or she subsequently develop typographic taste. However, they are indispensable if, as is becoming more common, you are printing your own proofs or preparing PostScript proof-files.

Another editing tool worth commenting on is the Find/Change tool. *FrameMaker*'s is more powerful than that of all but two other Mac programs I am familiar with. It can search for a wide variety of items, including plain text, specific character and paragraph formats, various special markers the program uses, and style-specific text copied to the clipboard. However, it can only replace with text (which takes on the attributes of the replaced text), with a character format (not restricted to ones in the catalog), or with the contents of the clipboard (preserving their attributes). One problem with this scheme is that since *FrameMaker* does not supplement the system clipboard with any of its own, searching for the contents of the clipboard and replacing with a different mixed-style string is impossible: *FrameMaker* needs at least two clipboards. But a program that aspires to be the ultimate tool for producers of long structured technical documents really needs more than this. Given any abstract string pattern ("regular expression") in

which specific attributes (typeface, font, etc.) are independently associated with individual elements of the pattern (characters, wildcards, special search variables), it should be possible to find all strings matching the pattern and replace them with the strings that would result if one were to leave the Find/Change window and operate on the found string, performing, say, a sequence of attribute changes, character changes holding attributes constant, reorderings, insertions and deletions, the same sequence of operations on each matching string. It is this kind of drudgery that computers are for, yet on the Mac, only *Nisus* and *Qued/M* have these capacities.

Another improvement that *FrameMaker* needs if it is to live up to its aspirations is an internal macro language. The *FrameMaker* documentation for the Mac helpfully suggests that users purchase a third-party macro program, but this is not an adequate solution. For example, very often one wants to make a sequence of passes through a piece of selected text, searching for one kind of element and changing it this way, then for another and changing it that way, and so on. However, at the completion of one pass the text becomes deselected. A macro for performing such a repetitive task would therefore have to reselect the text before the next pass, which means that before it starts, it would have to mark the beginning and end of the selected sequence and, after each pass, click and shift-click respectively on those markers. A built-in macro program can do this sort of thing without much difficulty, but while it is conceivable that *Tempo II* could manage the same for *FrameMaker*, I have yet to see how.

More positively, *FrameMaker* has very nice variable and cross-reference facilities. It comes with the usual collection of system variables, such as "Current Page #" and "Page Count." Inserting these in a master page header

or footer with appropriate text produces such identifiers as "Page 12 of 40." Other master-page variables will extract the text of a paragraph with a specified tag, or the closest such paragraph on the current body page. Header/footer variables are provided and you can give them your own definitions (see the earlier discussion of dictionary headers). For instance, if your paper titles are always in a single paragraph tagged "Title," then defining the variable "Header/Footer 1" as "<$paratext[Title]>" and putting this variable in the header on master pages, or right master pages, will result in the actual title of the paper appearing in the headers of the associated body pages and updating automatically if you change the title in the main text.

Two types of cross-reference are possible, page-relative and page-independent, and you can make cross-references from one file to another. An example of a page-relative cross-reference is "see my discussion of this and that on page 33." As you add and delete text in composing your document, your discussion of this and that may move around, but each time you update cross-references the correct page number will appear because you have inserted a marker at the beginning of your discussion and the cross-reference is tracking the marker. Of course, this kind of cross-reference is ultimately useful only if the distributed version of your document is produced by a process that either also tracks the markers or else doesn't change your page breaks. Page-independent cross-references are not so limited, and their most prominent use is to pick up the autonumbers of paragraphs.[4] For instance, if you have an autonumbered series of examples in your document, it partly defeats the purpose of autonumbering them if you type the numbers when referring to them in subsequent text—as in "compare (2.4) with the above"—for you will have to find and change these num-

bers by hand if the autonumbers change. The solution is to enter a cross-reference to the paragraph that is currently numbered 2.4, since if that number changes, all your cross-references to it will change correspondingly when you activate "Update cross-references."

A writer may well end up putting considerable effort into creating master pages; character, paragraph and cross-reference formats; variable definitions; and so on. Fortunately, it is easy to transport this sort of work across files. Once you have produced one document with a battery of customized settings, you can clear all text from it and save it as stationery (opening a stationery file doesn't open the actual file but creates a copy of it). Alternatively, if you have begun a new file or have opened a file created in a word processor and would now like to apply the settings from another *FrameMaker* document, you can import those settings, so long as your computer has enough memory to have the two documents open simultaneously.

Despite some limitations, then, *FrameMaker* possesses an array of tools for creating and formatting text that will attract many scholars. Those who

4. *FrameMaker*'s approximation of automatic endnotes involves typing them at the end of the document, giving each note a paragraph tag with an autonumber, then entering a cross-reference in the body of the text where you want the note number to appear. This allows automatic updating of note numbers when a new note is entered, but the order of the notes listed at the end can be wrong if text is moved around in the document. It also makes the process of converting footnotes to endnotes or vice-versa very laborious, a process that is accomplished with a click in *Word* or *Nisus*. The cross-referencing trick, elevated in the new *FrameMaker* manual to the status of official endnote method, is in fact the idea suggested in the faq (frequently asked questions) file for earlier versions of *FrameMaker* as a workaround for the program's *lack* of endnotes! (The *FrameMaker* faq file can be obtained by anonymous ftp from rtfm.mit.edu in the directory /pub/usenet/news.answers/frame. Anyone considering *FrameMaker* will find it a useful document to read through.) More positively, by using different autonumbering series, or this technique along with footnotes, one can have as many different series of notes in a single document as required.

already use a Mac word processor, however, are likely to wish that Frame Technology had made one small change in version 4 that would have eliminated much trivial aggravation. Unlike any other document-oriented Mac program I know of, when you open a file, *FrameMaker* puts you at its top instead of at the point where you were last working, and you have to scroll down searching for where that was. I am getting *used* to doing this, but my resentment at *having* to do it is undiminished.

GRAPHICS

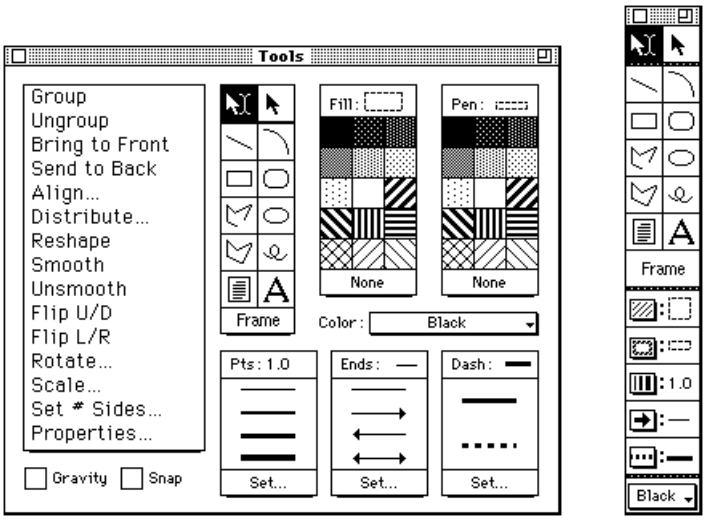

Figure 2: The graphics tools palette can be displayed in either wide or narrow view. In narrow view, the commands on the left in the wide view would be chosen from the Graphics menu. The highlighted tool is the Smart Selector, which toggles automatically between a text-entry cursor and an object manipulator according to what it is over.

FrameMaker has the best built-in drawing tools of any text-oriented Mac program. It lacks layers and graduated ("fountain") fills, but for relatively straightforward line diagrams, the need for a dedicated drawing package is eliminated. Specifically (see Figure 2) there are the usual line, arc, oval, and

rectangle tools, a Bézier freehand tool, and a useful "polyline" tool that allows one to draw fewer than *n* connected sides of an *n*-sided polygon as a single line (linguists will think of branching paths in a parse tree).

The Align and Distribute commands allow for precise relative positioning of drawn objects, while precise absolute positioning can be achieved either by editing the position properties of an object, or by magnifying the object and dragging it. The magnification range is 25%–1600%, and cursor keys are magnification-sensitive: an object can be shifted in increments of one point at 100% or one sixteenth of a point at 1600%. If your screen is big enough you can do all your writing and editing at a magnification in the 120%–200% range, which is very easy on the eyes.

Drawings can be placed in either an unanchored frame, which will remain where it is regardless of how surrounding text is edited, or an anchored frame, which keeps the same position relative to its local text, regardless of how the text moves, and can be copied or cut and pasted in a different location. There is considerable flexibility in the placement of an anchored frame relative to its anchor point, including the possibility of placing it completely or partially in the margin as well as in various parts of the text column. One limitation is that text cannot be made to flow around or alongside a frame in the text column, except by a tedious manual process that involves splitting the column and that has to be done all over again if the text moves. But arguably this is not a serious drawback, given the kinds of document that *FrameMaker* is intended to be used for.

A very strong feature of the program is the ability to incorporate text into drawings, either typed freely on a single line or entered in a text frame in the drawing. In *Nisus* and *Word*, the graphics modules have a text tool,

but a single text box cannot contain characters in different typefaces, fonts, or sizes. So a diagram label that mixes, say, Greek and English characters, has to be pieced together from different text boxes, with relatively unpredictable printing results. In *FrameMaker* (and also *WordPerfect 3*, though the latter lacks text on a line), text in a frame can be formatted just like text in the text columns of the document, and while paragraph formats cannot be applied to text on a line, character formats can be applied to the individual characters. Text on a line can be rotated in any increment, while rotation of frames is limited to 90° increments.

Graphics created in other programs, scans, clip art and so on, can be imported into a frame either by copying or by reference. Earlier versions of *FrameMaker* had a reputation for "losing" graphics imported by copying, but importing by reference is anyway a superior procedure: *FrameMaker* simply creates a pointer to the graphic file wherever it is on your disk, and puts a preview image into the frame. This keeps the *FrameMaker* file's size down, and if the graphic is one that you yourself have created in a graphics program, you can modify it in that program and the modifications will automatically appear in your *FrameMaker* file without your having to reimport the graphic. *FrameMaker* can import graphics in the following formats: Sun raster, EPSF (only the Mac version of *FrameMaker* displays the preview image), EPSI, EPS binary (only Mac and Windows *FrameMaker* display the preview image), TIFF, PCX, X11xwd, Xbitmap (UNIX *FrameMaker* only), MacPaint, RGD-SGI (UNIX *FrameMaker* only), DCS, GIF, PICT, and PostScript (the PostScript data is imported as text into a text column and the image prints correctly). The Windows *FrameMaker* supports some other formats, including BMP, CorelDraw, HPGL, and IGES.

Book Tools

If you are writing a book, you will want to take advantage of *FrameMaker*'s special bookbuilding capacities. Your first decision is whether to create the book in a number of files, say one file per chapter, or instead to put the main text of the book into a single file. *FrameMaker*'s manual recommends files of no more than 50 pages, on the grounds that opening and saving longer files is slow. But if you are using a Quadra-level Mac with a reasonably fast hard disk, you may well gain more than you lose by keeping to a single file. I have found that working with 200-page (≈ 800K) files is perfectly feasible. Although you do give up the option of chapter-relative page numbering (1.1, 1.2,…,2.1,…) since *FrameMaker* cannot reset page-numbering to 1 within a single file (conceivable workarounds with cross-references will defeat indexing), it is much more convenient in writing a book to have all one's text in a single file; for instance, cross-references are easier to create. And although *FrameMaker* provides many elaborate facilities for organizing chapter files into a book, some obvious ones are missing, such as the ability to find and change in all chapter files—if you don't discover until Chapter 8 that "Nietsche" is not how it is spelled, you will have to open each chapter file individually to make the change.

You organize your book by creating a "book file," which is a file that tracks the other files that make up the actual book. *FrameMaker* will then maintain page numbering and paragraph autonumbering from chapter to chapter according to your specifications. So, for example, if the page numbering is consecutive throughout the document (as opposed to chapter-relative) and you add some pages to an early chapter, *FrameMaker* automatically updates the starting page numbers of subsequent chapters.

Book-file organization also aids in the construction of indexes and tables of contents. A table of contents (toc) can be generated automatically by specifying the paragraph formats whose text and page numbers are to make up the toc. For this to work, of course, you must have consistently used the same paragraph tags in all chapter files for chapter titles and for section heads (overrides are permitted, so, e.g., spread adjustments can be made). Once you have set up such formats for one chapter you can ensure consistency simply by importing them into subsequent ones. Indexing requires that you manually assign a marker to each item to be indexed. The marker text can simply be the highlighted text in the document, or it can be other text with various formatting codes embedded in it, allowing an index entry to have subentries, to specify page ranges, to have special style attributes, and so on. Each index uses the same marker type, and *FrameMaker* comes supplied with a variety of named markers such as "Index," "Subject" and "Author," which allows one to construct many different indexes for the same document. Unfortunately, indexing is not well-integrated with Find/Change. For instance, it is not possible to search for all occurrences of an author's name in a chapter and assign them all the "Author" marker—one must search for them one by one, switching between the Find/Replace window and the Marker window. And to search for the first occurrence of some technical term in a book, one must open each file individually: there is no "search all book files" option. This is another point at which Frame Technology has missed glaringly obvious opportunities fully to automate repetitive tasks. Indexing requires patience, then, though the final results can be very professional. A nice touch, for yourself or another *FrameMaker* user to whom you are distributing your

document, is that when the program generates indexes and tables of contents, it can automatically insert hypertext links so that clicking on an item causes *FrameMaker* to display the page it references. You can also insert other hypertext links within your document, allowing a reader to jump from one part to another, related, part.[5]

NEW FEATURES IN *FRAMEMAKER 4*

Whether or not *FrameMaker 4*'s new features excite you depends very much on what you use the program for. Those who use it to create online documentation will appreciate the capacity to embed *QuickTime* movies, but this feature is of little use to those who are mainly concerned with producing camera-ready copy. Some other new features deliver somewhat less than they promise. For example, *FrameMaker* now supports System 7's Publish and Subscribe, but essentially only for formatted text from other Publish-capable word processors (which Frame would prefer you to abandon anyway). This reviewer wishes the effort had been put into fixing note-handling instead. *FrameMaker 4* also supports the XTND format translation

5. In *FrameMaker 4.0.2*, hypertext links are preserved when printing with Adobe's *Acrobat Distiller* as the selected driver. Frame Technology makes a separate application called *FrameReader* that can be used to read, move around in, and attach notes to, *FrameMaker* documents. Unfortunately, *FrameReader* licenses are relatively expensive, and the program assumes that the reading computer has the document's fonts installed in its system. Though it loses hypertext links, I think that the most cost-effective way of electronically distributing a *FrameMaker* document is in *Common Ground*'s "digital paper" format. A *Common Ground* file can embed a free miniviewer and gives a faithful reproduction of the original document even if the fonts used in it are not installed; and *Common Ground* works with other applications. *Common Ground* is available for the Mac and for Windows, with UNIX versions in preparation.

mechanism, but as the release notes on the installation disks inform you, useful XTND translators did not make it into the package. Version 4.0.2 does include a Mac and PC *Word* importer as well as a PC *WordPerfect 5.x* exporter and an RTF exporter. But to get the full benefit of XTND support, one has to buy a third-party translator library, typically either *Word for Word* or *MacLink Plus Translators*. These packages already include translators that will convert all Mac and most PC word processor formats into MIF (Maker Interchange Format), *FrameMaker*'s special interchange language (because Frame Technology will not release the details of *FrameMaker*'s native file format, many operations, especially involving third-party products, have to detour through MIF). In other words, if you buy *WfW* or *MLPT* to exploit XTND, you get equivalent functionality over again. To be sure, if you have the translators then *FrameMaker 4* can open files in other formats without detouring through MIF, but experimentation reveals that sometimes the MIF route produces better results. And whether interpreting MIF, using XTND translators, or using its built-in *Word* reader, I find that *FrameMaker* persistently mishandles certain fonts outside the standard LaserWriter set; for example, a switch to italic for one character in the original document will often produce italic to the end of the paragraph in the document that *FrameMaker* creates.[6] Finally, another important innovation in *FrameMaker 4* has also been underimplemented: only the minimum four AppleEvents are supported. This is a pity since AppleScript in conjunction with the Front-Most interface builder that ships with it will probably move into the mainstream as an environment customization tool having a lot of appeal for the

6. Unexpected stretches of italic also appear in files interpreted from the MIF files created by the bibliography program *EndNote*.

kind of sophisticated user to whom *FrameMaker* also appeals. Fortunately, most of the issues I have just detailed here can be addressed in the next release of the program.

FrameMaker 4 has much better color-handling than earlier versions, though again this feature may be useful only to a small subset of users. But perhaps if you give them color they will come. There are also various interface improvements and a new Utilities facility that provides word count and detailed document comparisons, useful if you are one of a number of co-authors of a document swapping drafts back and forth. Multiple foreign languages in a document receives support as well, though as far as I can tell it falls short of the WorldScript support to be found in *Nisus*.

The improvements to the program come at certain costs. The application's size has ballooned from *FrameMaker 3*'s 580KB to 2.8MB, bad news if, like me, you run *FrameMaker* from a RAM disk. Its memory requirements are also more demanding: it requires a RAM partition of at least 3MB and Frame recommends 5MB (I have found that an 800K file can be handled in a 3.8MB partition under System 6.0.7). Version 4.0 was also significantly slower than version 3.0. Fortunately, 4.0.2 seems to have corrected this.

Besides upgrades, there is now a new way in which capacities can be added to the program. Frame Technology has created a Developer's Tool Kit which will allow third-party programmers to write Extensions, creating a market like that of the XTensions for *QuarkXPress*. The danger is that Frame might be tempted to leave to third-party developers matters which are really its responsibility. For example, it has already been suggested in the magazine *Framers forum* that Extensions should be written for automatic endnotes and drop caps. But given Frame Technology's positioning of its

product, these should be features the company builds in, not extras it expects users to pay for on top of the cost of the program. Other functions, like database publishing and irregular text wrap round graphics, being rather more peripheral to the program's central purpose, could reasonably be left to the Extensions market, though I hope the exorbitant prices commonly charged for XTensions will not be imitated.[7]

Other Significant Features

I conclude this review by briefly mentioning three other significant features of *FrameMaker:* (a) tables, (b) equations, and (c) conditional text.

(a) *FrameMaker* has a full-featured built-in table editor. Tables can be autonumbered, can span pages, and can contain footnotes in their cells. Paragraph formats can be set up for cells (recall that "Table Cell" is one of the property sets in the paragraph designer), and cells can contain other tables or graphics. Custom shading can be applied to rows or columns of cells, and custom styles can be applied to borders and dividers. You can resize the width of a table or of one or more individual columns as you work, while *FrameMaker* automatically adjusts the height of each row to accommodate its contents. Both rows and columns can be moved around, and the whole table can be cut or copied and pasted in another location. In fact, in creating tables in *FrameMaker* I have not come across any real prob-

7. There is of course a thriving Mac market for system extensions. Some of these extensions have particular difficulty with *FrameMaker*. For example, the excellent *PopChar* does not display any fonts other than the standard LaserWriter set, and if *FrameMaker* is not on the exclusion list of *WYSIWYG Menus* (a component of *Now Utilities*) *FrameMaker*'s font menu will display only the standard LaserWriter fonts (though any others you have installed are accessible in the two designers).

lems.

(b) *FrameMaker 4* also has a built-in equation-editor, much improved from *FrameMaker 3*. Equations can be in-line or displayed, and the equations palette has eight pages of math elements, to which you can add your own, together with micro-positioning tools and useful on-line context-sensitive help.

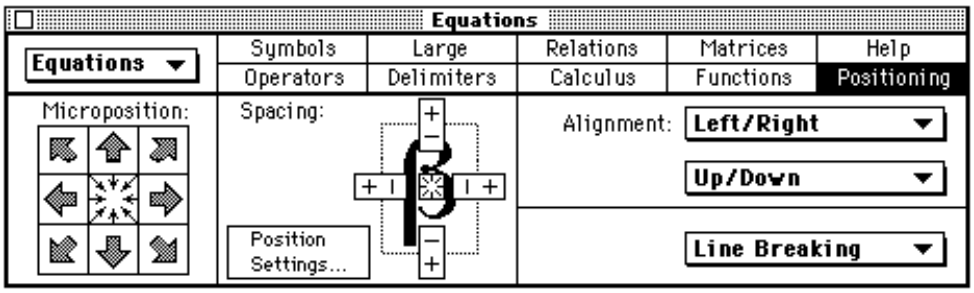

Figure 3: The equations palette has ten pages. Here the Positioning page is displayed. The drop-down Equations menu contains other formatting choices and gives you the ability to add new math elements, e.g., a new relation symbol.

The default font from which the math elements are drawn is the standard and somewhat limited Symbol font. According to the documentation, Adobe's Mathematical Pi fonts and Y&Y's Lucida NewMath fonts are also supported, but the feature does not work and Frame intends to fix it in the next release. Still, scholars in the formal sciences who would like the power of TeX without the pain should give *FrameMaker* serious consideration. There are filters which convert MIF into TeX and vice-versa, so journal submission requirements could still be met and colleagues' papers still opened. As well as formatting equations, *FrameMaker* can also *evaluate* equations. It isn't *Mathematica,* but yes, the program which cannot split a footnote across two pages can find the determinant of a matrix, divide out common factors, or calculate an integral.

(c) *FrameMaker* has an unusual conditional text and graphics capacity, which might be of use to academics who are, e.g, preparing student notes on the same basic material in different versions, say, for a regular and an honors section. Conditional text can also be used for non-printing notes. The procedure is to create a condition tag, say "honors," and apply it to text, frames, tables, footnotes, and whatever else is appropriate. The document will then have unconditional and conditional elements, and you can choose which combination to view, edit or print.

CONCLUSIONS

If only. If only *FrameMaker 4*'s note handling were on a par with that of the high-end Mac word processors, it would be the premier package available on the Mac, and probably under Windows and UNIX as well, for the kinds of document-creation needs that academics typically have: writing research papers, notes for class distribution, and composing camera-ready proofs for monographs and textbooks. But as things currently stand, scholars whose typical documents contain lengthy and numerous notes are faced with a dilemma. On the one hand, there is the large number of things that *FrameMaker* does very well, the same combination of which probably cannot be found anywhere else. On the other, there is the specter of footnotes showing up one or more pages adrift from their numbers (every time I made a change to the text of *Modern Logic* I had to make sure that it didn't knock a subsequent footnote in the chapter onto the wrong page), the hassle of checking that the sequence of endnotes reflects the sequence of cross-references to them in the document, and the time-consuming labor of manually converting footnotes to endnotes if a publisher asks for endnotes. My

own verdict is that the trade-off is worth it, but scholars in other disciplines or with other writing styles may not find it to be. It is high time for Frame Technology to abandon its casual attitude about this major flaw in its product.

A final comment, to end on the upbeat. Reading through this review it occurs to me that I may have created the impression that *FrameMaker* is a difficult program with a steep learning curve. This is not so. The Macintosh interface has been closely adhered to, so if you can already use *Nisus* or *Word* or *WordPerfect*—or even *MacWrite* or *WriteNow*—you know how to make basic use of *FrameMaker*. And the special features of the program seem to me to work in a logical way. For instance, I found it hardly necessary to read any of the manual's fifty-odd pages on equations to produce nicely formatted equations at the first try. Similarly, if you glance at the manual before creating your first autonumbered format or cross-reference, you should find the procedure on all subsequent occasions trivial. One thing that prevents people from switching to a new program of a particular kind is the slowdown in their work they anticipate while mastering the program. But on the Mac, in most of the major desktop software categories—word-processors, dtp programs, spreadsheets, drawing and painting programs—much of what you know about one program carries over to others in the same category, so you can get started straight away and learn the fine points as you go along. I found *FrameMaker* easily mastered in this way.[8]

8. Thanks to Carol Odlum for the many improvements she suggested to the first draft of this review.

Name of Program: *FrameMaker*

Version Reviewed: 4.0.2

Company Name: Frame Technology Corporation

Address: 1010 Rincon Circle, San Jose, CA 95131

Phone: 1-800-843-7263; 408-433-3311

Fax: 408-433-1928

International: Frame Technology International Limited, Unit 52, Airways Industrial Estate, Cloghran, Dublin 17, Ireland; 353-1-842-9566, (fax) -9478

System Requirements: Macintosh—any 68020 machine or better, minimum of 5MB of RAM, 8MB recommended, 8–18MB of disk space, System 6.0.7 or later; Windows PC—any 80386 machine or better, minimum of 8MB or RAM, 10–20MB of disk space, DOS 3.3 or later, Windows 3.1 or later; Sun—Sparc Workstation or compatible, minimum of 16MB of RAM (40MB swap space required), 20–90MB of disk space, Sun OS 4.1 and Solaris 2.1 or later, X Windows X11R4 or Open Windows 3; HP—9000 series machine, minimum of 16MB of RAM (40MB swap space required), 20–85MB of disk space, HP-UX 8.0 or later, X Windows X11R4 or X11R5; IBM—RS/6000, minimum of 16MB of RAM (40MB swap space required), 20–80MB of disk space, AIX 3.2.3, X Windows X11R4 or X11R5. (*FrameMaker 3* was not upgraded for the NeXT—apparently Frame is waiting to see if NEXTSTEP for Intel machines catches on.)

List Prices (single user): Mac and Windows, $895, UNIX, $1495. Academic price (Mac, single user), $195, with graduated reductions for multi-user licenses.

Printed Manuals: *Using FrameMaker* (≈ 550 pages), *Quick Reference* (58 pages), *Getting Started* (130 pages), and *Introducing FrameMaker 4* (32 pages).

Using FrameMaker is the best piece of Mac software documentation with which I am familiar.

Online Help: Excellent hypertext on-line help, available in context-sensitive mode. There are also a number of special online manuals, e.g., one detailing the new features of *FrameMaker 4*.

Technical Support: 408-922-2744, 7.00am–5.00pm Pacific time. There is a faxback service at 408-428-6153 for US and Canadian customers (begin by requesting the index of documents). There is Internet support at comments@frame.com (Canada, US, Pacific Rim, South America) and intltechsup@frame.com (all others); more online support in section 12 in the dtp vendors forum on Compuserve (go dtpvend); and Mac technical support on a dedicated BBS (see the release notes for *FrameMaker 4.0.2*). Technical notes, extra documentation, useful utilities, and so on, are available by anonymous ftp from ftp.frame.com. Registered users receive *FrameFootnotes* and *Frame Focus*, published by Frame Technology.

Other Support Resources: Frame Usenet newsgroup (comp.text.frame), Framers internet mailing list (majordomo@drd.com, no subject, message "subscribe framers *your e-mail address*"), *Framers forum* magazine (37213 SE Wildcat Mountain Drive, Eagle Creek, OR 97022-9696).

Graeme Forbes is Professor of Philosophy at Tulane University. His textbook Modern Logic *(Oxford University Press 1994) was laid out in* FrameMaker *and was written to be compatible with the logic-teaching program* MacLogic, *which he uses in his own classes. He can be contacted as forbes@mailhost.tcs.tulane.edu*