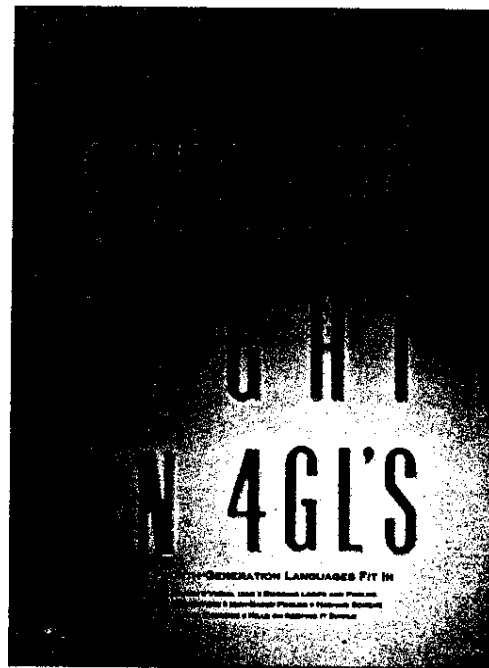


How to
Make

Desktop Publishing Work

'IEEE Software' makes the
successful transition to the
computer age

■ By Angela Burgess
and Galen Gruman



Desktop publishing is now well established in news-letter publishing, but can it handle magazines? Our company publishes six magazines totaling 3,800 editorial pages a year for a combined 150,000 subscribers, so the tantalizing promise of big savings demanded a close look.

When our publisher looked for candidates to explore desktop publishing, *IEEE Software* magazine was a logical choice. After all, the entire editorial staff (all two of us) is computer literate; Gruman had even reviewed the MS-DOS packages for *IEEE Software* and the trade newspaper *Infoworld*.

Still, we were skeptical. We'd been exposed to as much high-tech hype as

anyone. Our perception was that desktop publishing documents were of inferior quality and hard to produce. Besides, our artist's computer experience began and ended with his bank's automated teller machines.

Our preliminary analysis showed that because desktop publishing combined layout, design, typesetting, and production, it would be difficult to apply to our magazine. Sure, it's great for a one-person newsletter staff, but in our environment, job skills and duties are divided among several professionals. Desktop publishing had to work where editors edit and designers design.

But further investigation convinced us that desktop publishing *could* be

Continued on page 50

Desktop

applied to real magazines, at least to trade magazines—and it could save time and money without hurting quality or increasing anyone's workload.

We took delivery on our hardware and software in February 1988. Our May issue (which shipped in early April) carried 85 editorial pages created with Xerox Ventura Publisher.

In the process of learning to use desktop publishing, we had fun, stretched our minds, strengthened our working relationship, found a great new type vendor, and cut our total production costs by about 30 percent. We happily plugged part of our savings back into important things like design, writing, and art. We'll never go back.

We hadn't planned to move so quickly; in fact, we were cautiously predicting conversion by early 1989. What made the difference? Planning. The key to desktop publishing is not the technology but its proper implementation and management.

So, if you are an artist or editor who is unhappily resigned to learn more

about computers and technology than you ever wanted to know, take heart. There is a way to make the technology work for you—to make desktop technology adapt to your editorial process, not the other way around. Unfortunately, you will have to beat your own path for much of the way because every magazine is different. But we do have some nuggets of experience to offer.

Is it right for you?

We do not believe that this technology—which is, after all, only four years old—can handle every magazine equally well. To see if your magazine qualifies, ask yourself these questions:

- Is your magazine design-intensive, or do you follow a strict grid? Desktop-publishing programs don't encourage free-form experimentation. When our artist has to do a special spread, he still prefers working on paper. But most of our pages follow a straightforward format issue to

issue. We rarely wrap type around irregular shapes; we don't use special typefaces.

- Is your magazine automated? Do your editors already code and transmit type electronically from a word-processing program like Microsoft Word or Word Perfect? If you do, you can adapt your coding conventions to desktop publishing codes. If you don't, you're probably not ready.

- How much lead time do you have? Obviously, it's easier to convert if you publish a bimonthly than a monthly, a monthly than a weekly.

- Are you shipping on time? You'll need to invest time up front to save time (and money) down the road. If you never come up for air, you're going to have a rougher time converting.

- Do you have data-processing support? Or an editor on your staff who likes computers—an early adopter who can serve as a spark plug for the rest of your staff?

Without any one of these elements, converting your magazine is going to be harder than it was for us, but it is still possible.

CHOICES: SOFTWARE, HARDWARE, OUTPUT

Despite the insistent focus on computer-aided publishing, for most trade magazines the technology is neither the problem nor the solution. We produce about 100 editorial pages bimonthly for 25,000 subscribers. Our articles include many sidebars, tables, and figures. We use four-color and spot color on cover articles and special articles. Although we use original artwork and photographs, our design relies heavily on typography. We also use many special symbols and typographic effects in our copy because of its technical nature.

The current crop of desktop-publishing software can handle all these elements, and future upgrades will make some things easier to do; but we could get along fine with current technology for the foreseeable future.

Software. Before you choose a program, test the many ones available. Each has its strengths and weaknesses. None is perfect. On MS-DOS (IBM-

compatible) computers, the high-end programs include Xerox Ventura Publisher, Aldus PageMaker, and Laser Friendly Office Publisher. On Apple Macintosh computers, programs include Aldus PageMaker, Letraset Ready,Set,Go!, and QuarkX-Press.

Do more than look for a list of features. Try to do an article from start to finish so you can see where problems may arise and how the programs affect your workflow. The one best suited to your needs is something you'll uncover while test-driving them. We chose Ventura Publisher as best suited to our needs. The feature most important to us was the ability to code documents while editing, rather than transfer the mark-up (type specs) responsibilities to the artist. At the time, only Ventura Publisher offered this; PageMaker's new version also does, and a forthcoming new version of XPress is promised to do so.

Hardware. Hardware really isn't an issue unless you have no computers. Four years ago, only Apple Macintosh had the graphics features to handle desktop publishing well. The first desktop-publishing program, Studio Software's Front Page, was designed for MS-DOS (IBM-compatible) computers but was too slow and awkward to succeed. The Macintosh monopoly ended two years ago with the introduction of fast MS-DOS machines based on the Intel 80286 and 80386 chips (called 286 and 386 machines) with graphics adapters (Hercules and EGA cards).

Neither computer is superior to the other any longer. The main considerations, whether you want a Macintosh or MS-DOS machine, are screen size and resolution, hard-disk storage (40MB should be your minimum), and processing speed (10 megahertz should be your minimum).

Our best advice is to use whatever you have now. If you are editing on Macintosh computers, buy a powerful

Goals and requirements

We set two main goals: 1) adopting desktop publishing must save a significant amount of money—at least 30 percent—to be worth its use; 2) we must be allowed to retain the services of our artist, Kevin Reagan. An independent contractor, Reagan had been with us for three years, had overseen our redesign, and was an integral member of our team.

We also expected that 1) the editorial, design, and production effort should be no more difficult than under the current processes; 2) the magazine's appearance and quality must not suffer; and 3) we would be allowed (within reason) to make our own decisions about software, hardware, and vendors. Thankfully, our publisher was willing to give us a lot of control (or rope, depending on the result).

We first examined the costs (in time and money) of putting our publication together, from receiving copy to stripping negatives. These costs were easy

to quantify because we use outside vendors.

The cost analysis was done by winnowing out *only* those cost categories that would be affected by desktop publishing: layout of standard-format pages (not special spreads or extra design work), straight pasteup, most of the strips (because proofing is transferred to laser-printer output), and type.

To project savings, Burgess had to gather data on each category, interrogate Gruman (who was learning the software) on how desktop publishing could be applied, and make some common-sense judgments. That done, Burgess added a 10-percent fudge factor. Even then, she was able to project a savings of about 35 percent. The cost of type alone could be expected to be cut in half!

Analyze the technology

While Burgess worked on setting savings goals, Gruman learned Ventura inside and out. His exper-

tise was an obvious advantage, but the point is that one person on your staff—preferably someone who is most familiar with the production cycle—be designated to learn the software, perhaps in tandem with someone on your data-processing staff. This means freeing that person up, perhaps arranging for some specialized training.

Gruman soon realized that the weakest part of the program was that the software's front end (the editorial prep) was too labor-intensive. Without getting into too much detail, Ventura (and all other software packages) require that you break your copy into separate chunks if you want to manipulate them independently. It also requires you to assign a tag to each chunk, specifying the typeface, size, leading, width, and other typographic attributes.

This is too labor-intensive. Gruman's solution is a program, written in Basic, that reads our current typesetting codes and breaks the article into these separate chunks (for example, a

Continued on page 52

Macintosh II (with a large monitor and large hard disk) for your publishing workstation. If you are editing on MS-DOS computers, buy a fast 286-based machine (with a large monitor and large hard disk). If you have no computers, get ones that run the software you like best; the software should always be your deciding factor.

Prices range from \$3,500 for systems based on a 286 or Macintosh SE to \$5,500 for those on a Macintosh II to \$8,000 for those on a 386. In the MS-DOS world, you pay a premium of \$1,000 or so to have a major-brand label (like IBM and Compaq) on the hardware; such a name guarantees nothing about quality, power, or value. (No one but Apple sells the Macintosh, so there's no such price range.) Laser printers are another \$2,500 to \$4,000.

Whatever system you buy, we strongly recommend that you buy a two-page monitor. Available only since late 1987, these let you see a full spread (11x17 inches) on the screen at

once and read all type set at six points or larger. They are not cheap (about \$2,500), but they save a lot of time zooming between views where you can read text but see only part of a column and views where you can see the whole spread or page but not be able to read the text.

Full-page (vertical) monitors are appropriate only for design on single pages, such as ads. Most layout is done with spreads in mind, so full-page monitors are a wasted investment for most publications.

Output. We had no choice but to change some things, of course. Our type vendor didn't have a Linotronic typesetter, so we had to go looking for one. This was a matter of calling every type vendor in the area and asking a series of questions:

1. Do they have a PostScript-compatible typesetter (a Linotronic 100, 300, or 500 series)?

2. Do they have a sufficient font selection to handle house ads, cover type, and other special type jobs that

won't necessarily come in a PostScript format? This is an important question if you're considering a vendor that is not a traditional typesetter.

3. Is their bulletin board on-line 24 hours a day? Does transmission speed matter to them?

4. Is delivery included in the price?

5. Is it significantly cheaper (half as much) than the cost of modeming an equivalent amount of type and pasting up the type after layout?

6. Is the typesetter close enough to your office in case of emergency?

After visiting some sites, we selected a traditional typesetter who was new to desktop publishing over a computer expert who was new to typesetting. We figured that the process of getting type output from PostScript to a Linotronic is strictly mechanical, while the process of learning type is not. It was clear to us which experience was more important. Choosing a type house also meant we had access to full typesetting services for ads, covers, and other special work.

Desktop

file that contains the headline is assigned a ".hed" extension, a byline file gets a ".byl" extension) and inserts the "tags" in each file automatically.

If you don't have this kind of data-processing support, you can replace your codes with Ventura's as you edit, or globally with the search-and-replace feature in your word processor. That done, you can break the file manually into the chunks. Not as elegant, but serviceable.

Take the time at this stage to invest in typographic issues. It is not a good idea to rely on the default type settings provided by these packages. This is a good time to involve your candidate typesetters in the conversion. Have them look at some sample output from your laser printer and make suggestions to improve the appearance of your type. If you're looking for a new type vendor as part of your conversion, this is a good way to gauge their expertise and see if their standards for quality match yours.

The hyphenation dictionaries that come with desktop publishing programs are insufficient. To test them, take a sample article and lay it out in narrow columns (like eight picas) with justification and unlimited hyphenation on. This will force all the bad breaks you will likely encounter. Then add the corrected hyphenation to your exception dictionary (the better programs all have one).

We had to add 284 words to our dictionary, only half of which were technical terms. The others were common words like "quality," which Ventura hyphenates as "qu-ality." We also overrode some proper but rare hyphenations (for example, we overrode Ventura's hyphenation of "pro-ject," as a verb, to "proj-ect," the noun.)

We also adjusted the tracking (inter-letter spacing) and word spacing for our paragraph tags. All desktop publishing programs seem to use loose spacing unless you override them.

Target your training

Don't drag your whole staff along this learning curve. Pick a leader who can be counted on to be enthusiastic and helpful, someone who can

JALICS.DOC
Misra/Jalics
IEEE Software
July 1988

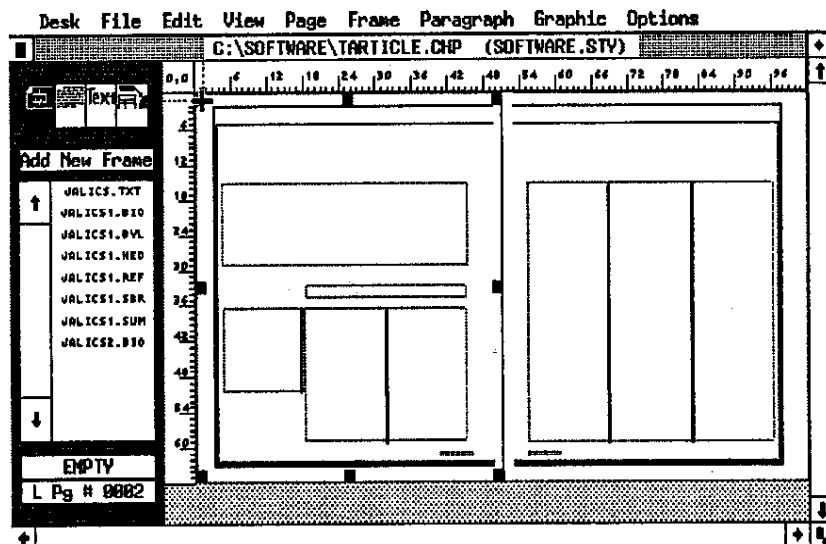
[ds1]Fourth-generation languages are not always better than their predecessors. This case study reveals where they do well and where they come up short.

[hs0]Third-Generation versus Fourth-Generation Software Development

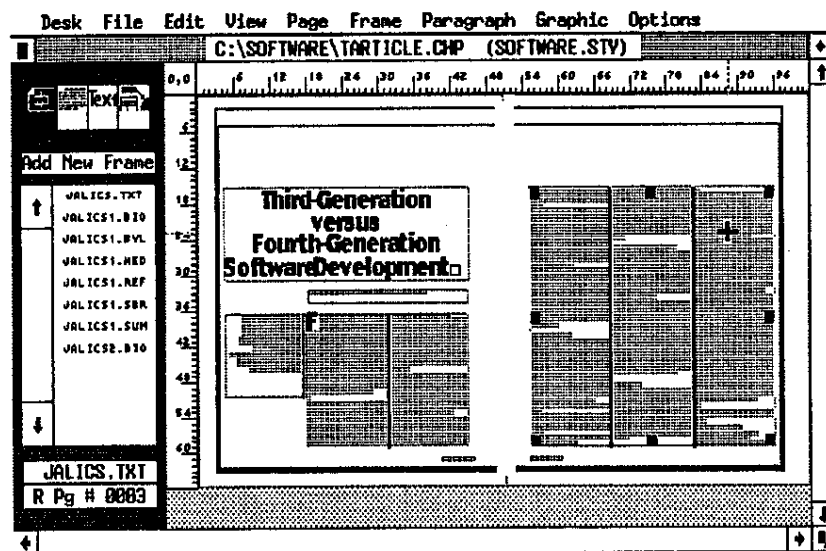
[bs1]Santosh K. Misra and Paul J. Jalics, Cleveland State University

[ks1]Fourth-generation language tools have made impressive gains in productivity, as several reports in recent years

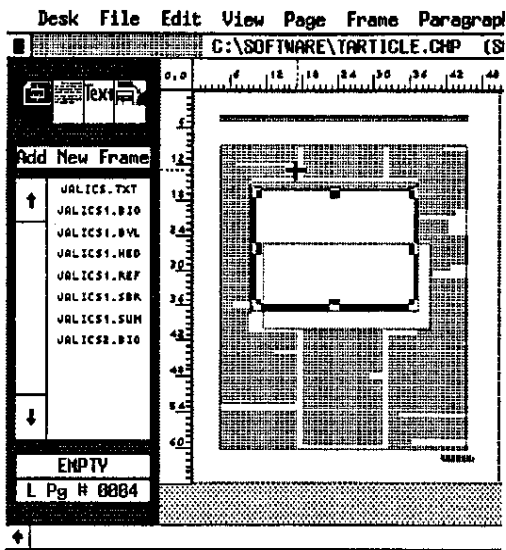
Word-processing file with embedded codes. The custom translation program breaks the document into the set of files that Ventura needs; [ds1] creates the summary deck, [hs0] the headline, [bs1] the byline, and [ks1] starts the article with a drop cap.



The key to Ventura Publisher is setting up a template—a layout grid. You then import the appropriate text files into each frame. The files for this article are listed at left.



The opening spread after the text files have been flowed into their frames.



Ventura's anchor feature lets you mark in your text where figures, tables, and the like should be. By anchoring the page, Ventura brings the frames that will hold the figures and tables onto the page that referenced them. To make this work well, you should size the figure and table frames before anchoring them. Once called to the page, you must still place them according to what looks best.

answer your staff's questions about how the software works as it pertains to your magazine.

Burgess was neither interested in becoming a power user, nor did she think it necessary to turn Reagan into a power user. This attitude helped target the training effort to those parts of the software that directly applied. Learning the rest of this complicated software can wait.

The hardware, software, and vendor in place, we were set to begin training. Our publisher questioned the wisdom of training an independent contractor (Reagan), but we argued that we would have to train someone one way or another. Either we trained the artist who knew the magazine on the technology or we trained an artist who knew the technology on our magazine.

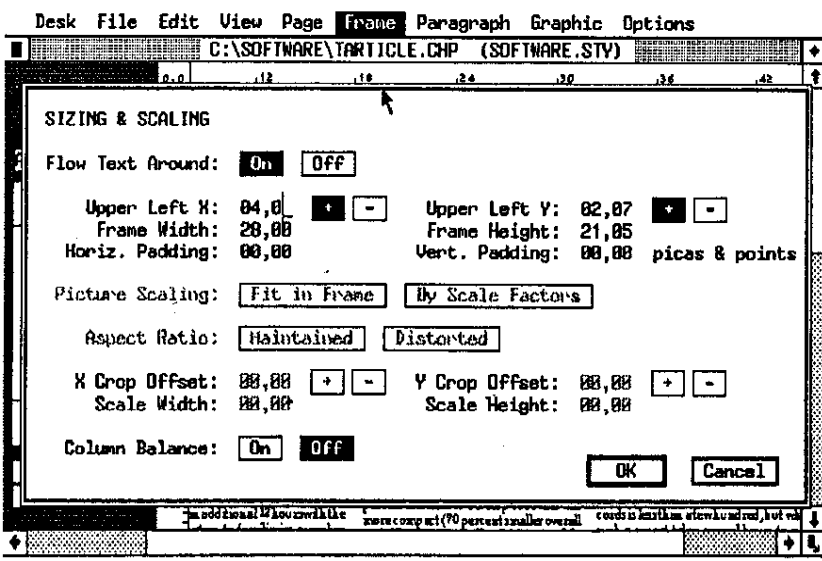
We resisted the use of outside trainers for the same reason: A total stranger would not know what we needed to know and what we didn't. Again, outside (read expensive) trainers don't know our magazine, so they could not target the training.

If you choose to use outside trainers, make them tailor their training to your needs and their expertise to an actual job from your magazine. Don't accept just a quick tour of the system's features, and don't expect much help from the manual for questions that you didn't ask during training.

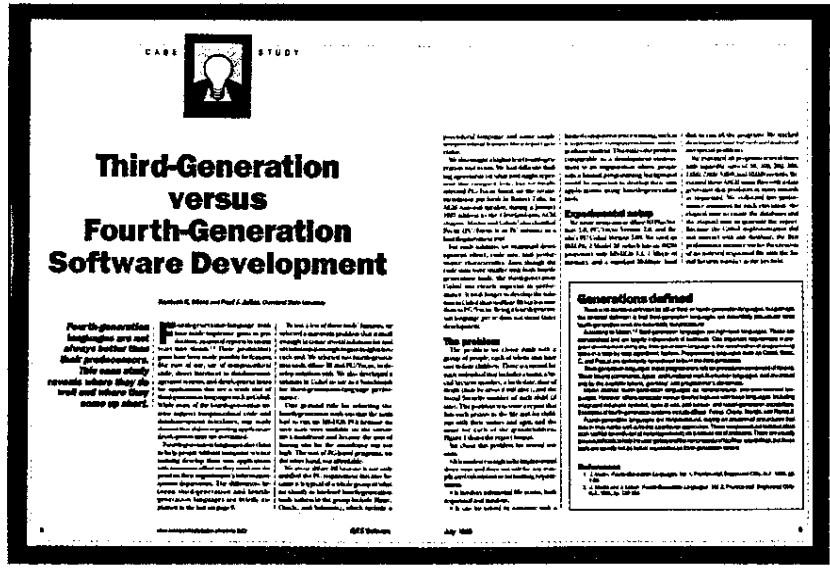
We began Reagan's training by having him help set up the style sheets and templates. Ventura requires a significant amount of setup, which made for a good introduction to how computers worked. Reagan soon realized that computers are dumb and unforgiving if details are omitted. He got used to manipulating the mouse (something that threw us all initially). And, because we experienced hardware crashes, he learned the valuable lesson that you can never save your work too often.

After four days on the computer, we were relieved when Reagan reported that he was having fun. While he still expressed doubts about his ability to take in all the details, his positive perception of the technology boded well for the project.

Continued on page 54



This menu, which Burgess and Gruman call "move it with math," lets you precisely place frames by entering the coordinates. Once you've set up your grid, you will know the *x*-coordinate (horizontal position) for each column, so choosing the *x* coordinate is easy. But the *y* coordinate (vertical position) is trickier because it depends on where the frame looks and fits best. Our designer often uses "move it with math" for exact horizontal placement but still relies on using the mouse and eyeballing it for vertical placement.



Final opening spread.

The template, style sheet, and typographic tweaking done, we proceeded to do a dummy layout that accounted for every possibility. The idea was to test everything, from the front end to the final output. This is a crucial exercise because it points out important shortcomings early on. Again, a lot of investment up front in Ventura really pays off.

This test exposed the most serious layout problem: how to get the software to place illustrations on the appropriate page automatically, without reading the text on the screen! Again, Gruman's familiarity with the software paid off. He took the idea of using what Ventura calls "anchors" (which pull elements onto pages automatically when they see a marker in the text) and modified the process to our needs. We had only to place the markers in the text with a new code: [fa1] to anchor figure 1, [sa1] for sidebar 1, etc.

Strategies emerge

A layout strategy soon emerged. When Reagan pulls up a fresh template, the first thing he does is import figures, tables, and sidebars and adjust their sizes. If the artwork is to be dropped in mechanically, Reagan sizes an empty box.

That done, he begins loading in the body copy. After loading each page, he anchors it, which pulls the presized boxes and their text onto the page. He places the boxes approximately where he wants them, flows the next page, anchors, and so on. Of course, placing a figure on a page very often causes its anchor to be bumped to the next page. So when Reagan anchors, the figures sometimes move, too! He quickly learned to work in a mode that displayed a spread on the screen so he could make design decisions to override the anchor command.

At the layout stage, we asked Reagan to leave things fairly loose. Rough layouts are enough to tell us the length and overall appearance of the article and whether we have room to place partial ads on the last page. Reagan invests no time here to bottom out columns or to place elements precisely.

UPDATE

We produced our second issue (105 editorial pages), which went to press June 10, almost completely with Ventura Publisher: all editorial pages but the reader-service cards and the front cover. We also stripped in department logos manually since we hadn't finished creating them electronically. They are done now and will be used in our next issue, which goes to press August 8.

Despite our prediction, the second issue took no longer than the first, because our designer remembered a lot more than we expected him to—despite a month's absence from the computer. In fact, he remembered and learned enough to use the technology for other clients in our organization.

This second issue marked the first that we had electronically laid-out departments, which have the multiple-element layouts we cautioned against. These were less difficult than we had expected, although we still recommend that new users tackle such layouts while on deadline only *after* they are comfortable with the technology.

After all, the text is not finalized yet, so why waste the time and money to make its layout perfect?

Later, when the text has been proofed and corrected and the ad counts are in, the pasteup process can begin. In our new vernacular, pasteup means bottoming out columns, inserting body decks, electronically flopping an article when necessary, placing spaces for ads, and writing the file to the PostScript format for our typesetter. The only traditional pasteup that remains is putting down the full pages on the boards, cutting ambers for screens, keying in photos, placing line art, and spec'ing color.

Lessons learned

After producing one article with desktop publishing, we realized we could do them all—the learning effort was concentrated in that first article. Among the lessons we learned were the following:

- In three days, our designer laid

out nine articles, which took up 80 pages. At first, we looked over his shoulder and nudged him now and again (more than that, in his recollection!). Pasteup took another three days. We believe that the next issue may take a little longer because we won't nudge him so much, but future issues will take a little less time because he will be familiar with the system.

- We learned that multiple-element layouts, such as news and product departments, are the most difficult to handle—after all, even on paper, many elements are more difficult to lay out than a few elements. It is even more difficult with computers to juggle many elements to see what combination works best because computers are designed to work single-mindedly. Don't start your transfer

to desktop publishing with these kinds of layouts; wait until you are comfortable with more straightforward problems.

- We also realized that stripping in corrections is now more expensive because we pay for type by the page, not by the amount of type on that page. That means we must minimize strips after outputting the page. Proofing from laser printers helped us catch errors earlier. What we used to find after typesetting and pasteup we now find when proofing the laser-printer copy of the layout.

- We were unsure how desktop publishing would affect advertising production. Because desktop publishing deals with completed pages, we now send our completed pages to type two to three weeks later than we used to. Previously, we manually moved type around to accommodate ads. We feared desktop publishing might lock us in to a page imposition sooner than we wanted, costing us some flexibility in ad placement and color decisions. This fear was unfounded. If

need be, we can still cut the pages up and walk or remove type to accommodate late ads or we can fix the layout in Ventura and reset the whole page. What you do will depend on schedule and budget.

- Typesetting at a flat page rate also means that design-intensive magazines may not realize the savings we did. We fit about 900 words per page and had paid about \$23 per page for modemed, precoded type. If we ran only 450 words per page, we would have paid about \$12 per page for type. With desktop publishing, we pay \$8 per page no matter how many words are on the page.

- We learned that Ventura viewed our pages as grids with an x,y axis and how to use that to place items precisely and easily. Although a difficult concept to explain, this feature eliminates the frustrating ritual of trying to place elements with a mouse. It also means we don't have to rely on an imprecise on-screen ruler.

Once he learned to move elements with math, Reagan could place them precisely—better than he could in paper pasteup. In fact, the placement is so precise that our pages look almost machined. Other programs don't have such a feature, but they can display your current position as you move the mouse so you can see how close you are to where you want to be.

We are already looking ahead to bypassing boards altogether and outputting negatives. (PostScript typesetters can produce negatives for about \$15 per page.) That means *no* corrections allowed. It also means that our prep house will have little to do beyond assembling the cover, stripping in halftones and artwork, and doing the negatives for color.

Later this summer, we will investigate software packages for line art. Reagan now does all the technical figures with india ink. For us, it costs no more than what some of our company's magazines spend to produce fig-

ures with a program like MacDraw or Adobe Illustrator.

And in three or four years, perhaps there will be affordable scanners that can read in color graphics at 2540-dpi resolution (about the same as 133 lines) and automatically do separations (including turning the negatives). Then you'll be able to do with desktop publishing what prep houses now do. It's the next logical step.

Less than a year ago, we thought desktop publishing was a technology that was oversold for real publishing. It is oversold, but proper management can make the technology work, at least for trade magazines. It is certainly worth serious investigation. ■

Angela Burgess is the managing editor of IEEE Software, a bimonthly trade magazine for software engineers. Galen Gruman is the assistant editor of IEEE Software and a desktop-publishing reviewer for the computer-industry trade weekly Infoworld.



LOOKING FOR OLD PICTURES?
WOW 'EM WITH OUR
THREE MILLION!