Interleaf 5: A Complete Overhaul of TPS

This article presents an overview of Interleaf's announcement of its new Interleaf 5 technology October 3 at the Seybold Computer Publishing Conference. Although it looks much like TPS, Interleaf 5 is a radical departure for the company. Rather than merely add features to TPS, Interleaf has taken it apart and constructed a set of programmable modules that may be tied together in a variety of ways and may be owned by users or vendors. A central component of all of the modules is the ability to tailor any aspect of the system—even to program a document to do certain things without operator intervention, what Interleaf calls "active documents." In addition, several key changes to the underlying operating system engine will enable Interleaf to bring its software into alignment with contemporary operating environments. To publish the information in a timely fashion, we prepared this story in advance, based on visits to Interleaf.

Interleaf's Technical Publishing Software (TPS) has been the leading publishing software for Unix workstations for several years, but in the past year or so the company's stature has diminished. Its desktop program—Interleaf for PCs and Macintoshes—has been sold sluggishly and did not establish Interleaf in the mass market the way it had hoped. At the same time, the company has been under pressure to expand beyond Unix from selling the Unix system to software that were once its core business to selling just software and services, such as consulting, training and support.

Interleaf faced several problems. Firstly, its unique combination of a fast WYSIWYG test editor and superior graphics editors were no longer sufficient differentiators in the market. Its workstations software sales remained strong, but the competition increased. Interleaf's preeminence had been based on a Dell 210 PC (a TPS-260) and a QMS PS-410 PostScript printer. The same software could be run on any computer and could be used with Unix. Interleaf's new director of product marketing, Both Fears and Wex will report to Berntz.

Allan Aldorf has joined Mycro-Tek as vice president of finance and chief financial officer. He was previously corporate treasurer at Westrex Investments.

MID/Information Logistics Group has hired Pamela Gennessu as a consultant. She comes from Datalogics, for which MID is the European distributor. She will specialize in SGML and CALS applications. PAM is a member of the ANL and ISO working groups, which created and reviewed the standard for SGML. She currently chairs the DoD CALS Educational Publishing Committee.

AAPT offers new Chameleon proofer

Baby version supports Core

AAPT is offering a new, less expensive version of its Chameleon proofing system. Called the Baby Chameleon, it is used internally, a small software company owned by Prime. With Prime, she had been vice president of marketing and technical support, managing a 250-person department.

Eugene Amato

APT has one major feature that from which software applications may be built. The toolkit fits Interleaf's view of its role in the industry—it must broaden its services at the same time that it focuses on offering specific inside-the-computer customer problems. The toolkit also will enable Interleaf to offer a consistent base level of software across all of its platforms and, starting next year, to offer products that comply with other graphic user interfaces.

One of Interleaf 5's distinguishing traits is its support for "active documents." In rebuilding its software, Interleaf has made every component of a document, both text and graphics, an object that may have programmable attributes. We wrote about Interleaf's active documents concept when it was introduced at the Seybold Seminars last March (see Vol. 19, No. 14). With the introduction of Interleaf 5, this feature is now realized in a product.

In one sense, Interleaf 5 is Interleaf's seventh major release to TPS in eight years. From that point of view, Interleaf 5 is TPS 5.0—TPS as we know it with the addition of some programming extensions.

At first glance, there is little in Interleaf 5 to differentiate it from TPS 4.0. But a closer examination reveals fundamental, underlying differences in the new code. According to Steve Pelletier, Interleaf's chief technical officer, the guts of TPS were overhauled more in this release than in any other. There are three main areas in which the overhaul is significant:

- Modular architecture. Interleaf is calling Interleaf 5 an open architecture, but "open" means many different things to different people. In Interleaf 5, the architecture is a modular system that is programmable at every level, allowing it to run on a variety of hardware and software platforms.
and to be tightly coupled with many application programs. It is the first publishing product that was designed for integrating publishing-specific functions with other applications— all driven from a document model. And it is one of the TPS 4.0, although there are new features. Interleaf has solved most of the problems of combining text and graphics on the page, and it has done so in a way that end users will find fast and usable. Thus, it could be a compound document vector feature that it is known for are still present.

But by breaking the engine into a custom module. Interleaf now makes it possible to insert external code that executes on the way in or out of one of these modules. 

For example, one manufacturer is working on a hypenation module for Interleaf 5. It will plug into the system at the engine level, so that the user will see no differences in the way the system behaves, but the composition might make better breaks in the new module, which uses ranked preferential hyphenation policies.

Other extensions could be written to create a new configuration—whether it is a structured editor that keeps an author within the bounds of a predetermined outline and style, a full compound document editor with a user’s favorite keyboard mappings or a form processor, in which a document is used as the interface for accessing a database.

Great! But—LSIP? While the ability to extend a system and define new object attributes by writing your own code is a powerful feature, we wonder how many end users will feel comfortable with LSIP. The language has a rather peculiar syntax with relatively few programmers (and even fewer non-programmers) are familiar. It stands in sharp contrast to the scripting languages, such as those in HyperCard or Excel, that were designed for end-user programming. (Some cynics have even suggested that LSIP was designed to assure itself of lots of custom-programming business.) We suspect that users will experience some frustration in getting up to speed on LSIP, but the effort is well worth it.

Font resources. All previous versions of Interleaf software used a raster font model that provided extremely fast display but imposed a burden in disk space and limited most of its version to disk-based fonts. With Interleaf 5, Interleaf has moved to outline-based technology that adds new functionality and decreases the program’s footprint on the disk.

Initially, Bitstream’s Speedo will be built into all versions of Interleaf 5. The LaserWriter Plus set of PostScript fonts is the standard font set for all Interleaf 5 products, but all will also support any additional Bitstream’s font that the user might like.

The software supports arbitrary sizing of fonts to 10s of a point in sizes ranging from 4 to 400 points. In addition, the display of any document may be enlarged or reduced in any font size ranging from 20% to 100% of its original size.

For the first time, Interleaf will be offering display of facing pages. If you scale the page small enough, you get
The display and user interface. Ever since its first product (remember OPS), Interleaf’s trademark has been its iconic desktop and pop-up cascading menus. Introduced in 1983, this approach followed the Xerox Star but predated the Apple Macintosh and the new graphical user interfaces: Windows, OpenLook and Motif. When it introduced its Macintosh and PC Publisher products in 1987, Interleaf made a conscious decision to offer its customer base desktop products that were consistent in look and feel with its Unix-based products.

Today the marketplace has changed. It is now an accepted fact that graphical user interfaces will be part of the standard operating system, although in the Unix world there are still at least two major interfaces to contend with (Motif and OpenLook). Developers in all computer applications are being forced to adapt their programs to these graphical interfaces or face rejection in the market. For developers who are increasingly offering their products on multiple computers, the burden of adapting the interface to each environment is not trivial—unless it is planned for in the underlying code.

Interleaf’s answer will be to separate the window drawing and user interface aspects of its code completely from its underlying engine components. We are told, because this work will not be complete until 1991. In the first release, Interleaf 5 will make use of X Window for displaying windows, but it will not offer alternative user interfaces. For example, it demonstrated Interleaf 5 at the Conference running on Sun’s new OpenWindows, but it did not show support for Open Look, which is promised for 1991. This is a good example of how, on the surface, Interleaf looks much like TPS, but underneath there have been significant changes.

By next year, Interleaf will be making use of the interface toolkits provided by hardware vendors. The Sun Open Look and OSF/Motif versions of Interleaf 5 will utilize those toolkits and the Mac version will make use of Apple’s Macintosh Toolbox. In so doing, Interleaf has made it possible for itself to develop an Interleaf 5-based product that conforms to whatever graphical user interface is required.

That much is not that different from what Frame did several years ago. But Interleaf is taking it one step further. Because Interleaf 5’s interface layer is a set of programmable objects, it is possible to write LISP scripts that interact with the user interface independent of the underlying engine code. A VAR or sophisticated data object can actually tailor the user interface of an Interleaf 5 product without any source license, much as users can create database query forms with today’s database management products. To our knowledge, Interleaf is the first vendor to introduce such a user-configurable interface for a publishing product.

(Admittedly, it is more likely that Interleaf, a VAR or a customer system administrator would do this customization for end users, rather than the end users doing it themselves. This is, after all, programming in LISP, not HyperCard buttons. But in the academic in environment it is not unreasonable to expect that clever end users will indeed make such modifications themselves.)

Revision tracking. Now a standard engine feature, revision tracking traces edits to text and graphics; it tracks who made the edits, when and in what version of the document. Top: The attributes in the competitor’s property sheet. Bottom: The style of edit tracking is set in the property sheet of the document version.

One question that remains is how LISP routines written for one environment will act in another (the Mac Toolbox has different functions from the Open Look toolkit). Interleaf is exploring that interaction, but not until next year will we know what the user interface interface to the document. For example, Interleaf hot buttons—in-document icons that initiate actions (comparable to HyperCard buttons)—are portable across all of Interleaf 5’s platforms without any additional programming.

Interleaf 5 features

In addition to the fundamental changes described above, Interleaf has added a new set of publishing features to its core software. We’ll describe first the most dramatic changes, followed by a list of other noteworthy improvements.

Revision control. In TPS 4.0, it is possible to initiate edit trace to track different versions of a document, but the information that is tracked must be custom-designed by the end user or Interleaf. In Interleaf 5, revision control is a basic engine component that, when part of the package, appears in the menu bar at the top of the document with a complete dialog box listing all of the possible items to be tracked.

Through the property sheet, the user establishes the style for tracking different types of edits—strike-through, underlines, different colors or revision bars, for example. It is possible to have multiple editing sessions within a single version—each of which may have different styles—which allows different people to edit the same version and differentiate among each other’s edits.

The system automatically tracks edits to text and graphics (additions and deletions) that made the change, when, on what system and at what time. When edits are approved, a new version is saved, using a hierarchical numbering scheme. The popup menu allows the user to access any version, current or previous, at any time.

Using the Relational Document Manager (RDM), an Interleaf product based on Oracle, it would be possible to add further controls, such as restricting access depending on revision level. Interleaf does not offer Context’s feature of arbitrary display of different editing sessions (e.g., Bob’s edits but not Sally’s) unless they are recorded as separate versions, but this feature does remain all of the revisions as iterations of the same document, rather than storing them all as separate documents. Storing multiple versions as iterations of the same document makes it easier to trace the history of changes and also saves disk space.

In general, Interleaf’s new revision control is among the best, if not the best, in the market. A key difference between the revision tracking of Interleaf 5 and that of Context, which prides itself on this feature, is that Context stores the revision information inside the document. In complex applications, Interleaf tracks revisions in RDM (outside the document in an Oracle database) so that the document does not contain all of the changes. In addition, Interleaf offers revision control in its graphic object properties, while Context offers only non-graphic object properties to be tracked.

Graphical object masters. Another welcome improvement is the addition of a property sheet for all graphic objects created in Interleaf 5. In TPS 4.0, only the frame itself had a property sheet. In Interleaf 5, every diagramming object may be named, and every object, or group of objects, within a frame has its own property sheet. The naming of objects brings Interleaf’s graphics into its text model, where each paragraph has a name (what Interleaf calls components, or what is commonly known as a tag). By naming objects, Interleaf is able to apply the style-sheet approach to graphics as well as text.

For example, a drawing may be created in which one element is repeated numerous times. Later, when a change is made to one of those elements, it is now possible to update all of the others automatically to reflect the change, yet still maintain transforms such as stretch, shear or rotate (see photos). The change may be one of style (fills, colors, etc.), or it may be content (position, rotation, addition or deletion of objects, etc.). This feature is a powerful addition to an already-strong illustration system.
The real power of Interleaf 5 is that LISP code may be attached to objects as an attribute.

As if that weren't enough, there is an additional aspect to the named object, one that is shared by all objects (text, frames, pages and documents). TPS 4.0 supported the use of "extensible objects"—those with known data type—but it was not a feature available at the user interface level. This feature made it possible, for example, to include graphics in Interleaf documents, for which Interleaf had no filter, or to attach security levels to documents, but it did not let you attach scripts to objects. In Interleaf 5, the program doesn't care if attributes are data or LISP code, and Interleaf has brought the object attributes out to the user interface.

For the user, the ability to create attributes easily can be quite handy. An obvious application is assigning levels of security to objects and then selectively displaying or suppressing elements according to security level. But, in another example, an educator might create a test that has questions of different degrees of difficulty, with answers that are also dements according to their security level. But, in embedded in an output file. The attribute of a link to another document and scrolling to the link destination. ArborText has done this in a similar fashion in its current version of The Interactive equation editor.

Keyboard is completely user-modifiable in all Interleaf 5-based products, so you can map your own set of keystrokes to keyboard characters to any special or math characters, Interleaf 5 supports the use of conditional variables, but if that weren't enough, there is one additional aspect that the popup cascades across the item it anticipates you will want to pick, the palette is much faster and more in keeping with the interfaces of other application programs. Like most aspects of Interleaf 5, the palette may be customized by adding your own graphics to it.

Repeating column headers. Running heads and foots may now be different for columns as well as facing pages.

Multiple page-numbering streams. Developed specifically for the pharmaceutical market, this feature allows a single document to have more than one set of folios.

PostScript color output. The spot color facility of TPS 4.0, which created spot color separations, has been extended to allow the printing of all, or some, layers on PostScript color printers, such as the QMS ColorScript.

External references in images. Images may now be included by reference without pasting them into the document, a feature that saves disk space and is useful when images are created by someone other than the author.

Extended search and replace. The search and replace is now case sensitive and has a robust set of wildcards.

Authoring support. The keyboard is completely user-modifiable in all Interleaf 5-based products, so you can map your favorite actions to whatever key you like. The macro facility of Interleaf 5 enables authors to not only save keystrokes and mouse clicks to keyboard combinations but also to attach scripts to keys. XWrite has a similar capability, and we find it an invaluable time-saver.

Improved hypenation control. You now can specify the number of letters that may precede or follow a hyphen. The minimum number of letters in a hyphenated word remains fixed at 5.

Improved leading control. Leading may be specified as baseline to baseline, and you can also specify leading by setting the line of hyphenation.

Compatibility with other versions. Every software vendor offers some sort of migration path for converting documents created in old versions into new versions of their software, but oftentimes they neglect to offer a downward translation. We are glad to see that Interleaf has included a "convert to previous" feature that converts Interleaf 5 documents to TPS 4.0 or Interleaf 4.0 (or even Mac or Word). In addition, when converting documents from older versions to the new, the user has the option of freezing composition so that when opened in Interleaf 5, the document will retain exactly the same line endings and pagination.

Packaging a product

Unlike past Interleaf products, Interleaf 5 is not a single product. The core technology may be packaged in a variety of different ways to fit different requirements. Interleaf sees three types of end-user products: horizontal applications, which are aimed at a broad class of users much like TPS or Interleaf Publisher;

vertical-market applications, which are general packages tailored to specific industries and applications; and customer-specific ones, such as developed for Grumman. Horizontal applications will be sold primarily by the direct sales force, with help from deals with hardware manufacturers. The vertical market packages will be offered by VARs and by Interleaf sales people trained in that application. The customer-specific products are sold and developed by Interleaf's system integration division, which is currently the fastest-growing portion of the company. Where applicable, Interleaf develops vertical-market products as offshoots of custom work.

Horizontal applications. At the Conference, Interleaf showcased six general-purpose packages built from Interleaf 5. All rely on the core TPS engine, but with different capabilities that reflect their primary purposes:

• Passport. A general-purpose compound document editor designed for everyday use in the office. Passport is much like TPS but with the addition of the active-document technology and without some of the advanced features (tables, equations, gray-scale editing, revision control, etc.). Initially, it will be offered with Interleaf's user interface. Broader success may be more likely next year, when Interleaf introduces Open Look and Motif versions, pitting this Interleaf 5 product against FrameMaker, which is now available on more than 20 different Unix workstations. Passport will lack some of the features of FrameMaker, but it may offer some that Frame lacks, such as Interleaf's graphics and charting facilities.

• Interleaf Engineer. Similar to Passport, this package adds the table and equation editors and the Methods Toolkit (devised for engineering applications).

• Professional Writer. In addition to the features of Passport, this package offers revision tracking, the optional Houghton Mifflin Writer's Helpers (dashed boxes), hyperlinks, live links to Lotus 1-2-3 and other applications and configurable keyboards for emboldening other word processors.

• Interleaf Illustrator. Interleaf's graphics editors are no substitute for Auto-trol or InterCap systems, but they are better suited to creating non-technical drawings, such as flow-charts, viewgraphs and diagrams. Adobe has poured Illustrator into DEC workstations, but as yet the market for Unix general-purpose illustration tools is wide open. Interleaf's Illustrate is a full version of Interleaf's graphics capabilities—drawing, gray-scale and line-art image manipulation, graphic text, the new graphic style sheets and a palette interface as an alternative to the popup cascading menus.

• Interleaf Production. This is the full-blown package that includes all of Interleaf 5's functionality. As with the other packages, it may be extended and tailored through the use of LISP scripts.

• Interleaf Academic. Containing all of Interleaf Production, plus the Methods Toolkit, this package is available free to accredited colleges and universities in the U.S. and Canada. All of these products may be extended by LISP scripts. Those customers who buy a product with a toolkit will have complete access to all of the software's components. Those who buy a subset of the engine may upgrade to receive more functionality.

Interleaf expects all of these packages to be available by the end of the year. When we prepared this report, Interleaf was still finalizing the pricing. We expect it to range from about $2,400 to $16,000, but we will report the actual numbers as they are made available.
Interleaf developed a product in which the entire user interface that the engineer deals with for an ECO is a form that he fills out. The Interleaf 5-based product uses Interleaf's Relational Document Manager to generate an active Interleaf document from fields in the RDM database. Because of its ability to attach specific documents to objects, Interleaf is able to do such things as automatically fill in certain fields of the ECO when others are entered (enter part number, get part name), verify the validity of certain fields (entered value must meet certain criteria or user is prompted whether value is correct), and automatically route the ECO when it is finished. When an order is placed, the same document can find and place in itself a digitized signature when someone in the chain gives approval. Yet the interface for everyone who interacts with the ECO is a self-contained form that is a document (see photo). A similar application is the job instruction card, which tells the maintenance person what specific tasks are required in a maintenance procedure. Today, some firms create these task cards by photocopying parts of the manual and pasting them together into an instruction card. In Interleaf's software, the maintenance information from which instructions are extracted is kept as a catalog of data files that is tracked in RDM. The header fields that identify the card are also tracked in Oracle. The software automatically retrieves the header information and maintenance instructions, based on the task number(s).

Critical applications. Interleaf 5 also makes possible the building of a special product for niche markets, and Interleaf has developed ones for aerospace and for Amoco, which will be used in petroleum engineering. (We described its work with Amoco in our coverage from Seybold Seminars, Vol. 19, No. 5.) In addition to these markets, Interleaf announced its intention to develop vertical applications for pharmaceuticals, computers, and automobile manufacturers.

Of these, the aerospace market is getting the most attention. Interleaf has set up a business unit, led by Larry Bohn, specifically to address contracts that demand compliance with the guidelines of the Air Transport Association (ATA, the airlines) and the Aerospace Industry Association (AIA, the manufacturers). The unit has subcontracted to 22 people in just six months, and not just in anticipation of orders. Interleaf has already won contracts with America West, Eastern, and, most recently, TWA. It also has won contracts with suppliers: Canadair, Boeing, Grumman, and Saab.

ATA demo. At the conference, Interleaf demonstrated an application developed for an airline and was "productized" for use at other airlines and aircraft suppliers. The application helps firms automate the process of producing documents that comply with the Airline Transport Association (ATA) guidelines.

The first application is an engineering change order (ECO), a document that is created in response to changes in equipment. Typically, the manufacturer or airline has one person whose job is to obtain approvals for the ECO and coordinate any requested changes. The approval cycle usually involves several people from differing departments. The change order must be routed (typically in hard copy) to various people for notification, approval or action, and the coordinator compiles all of the changes into the new release of the ECO.

Interleaf's RDM database contains the entire ECO, which is a self-contained form that is a document. It poses the user interface, which is a form that the engineer fills out. The Interleaf 5-based product uses the Relational Document Manager to generate an active Interleaf document from fields in the RDM database. Because of its ability to attach specific documents to objects, Interleaf is able to do such things as automatically fill in certain fields of the ECO when others are entered (enter part number, get part name), verify the validity of certain fields (entered value must meet certain criteria or user is prompted whether value is correct), and automatically route the ECO when it is finished. When an order is placed, the same document can find and place in itself a digitized signature when someone in the chain gives approval. Yet the interface for everyone who interacts with the ECO is a self-contained form that is a document (see photo).

A similar application is the job instruction card, which tells the maintenance person what specific tasks are required in a maintenance procedure. Today, some firms create these task cards by photocopying parts of the manual and pasting them together into an instruction card. In Interleaf's software, the maintenance information from which instructions are extracted is kept as a catalog of data files that is tracked in RDM. The header fields that identify the card are also tracked in Oracle. The software automatically retrieves the header information and maintenance instructions, based on the task number(s).

Graphics associated with tasks are automatically made available for placement within the card. Thus, the entire process is automated, resulting in a compound electronic document that may be made available as a file attachment to the card, or as a hard copy of the card that has been printed. Interleaf also has developed an aerospace-specific version of Professional Writer to help authors create aircraft maintenance information manuals. The manuals supplied by the manufacturer; others have the manufacturer do it for them under contract. The manual itself describes all of the maintenance procedures, following the naming and numbering conventions established by the airline industry. Interleaf took these names and numbers and added them to the standard menu, so that when a writer begins a new task, it may be created by picking its part number or description from the menu. The software then automatically inserts both the item selected and the item's part number, checks its validity through RDM and inserts it into the document. Although it was not shown, presumably it would be possible to further constrain the menu to show only the appropriate subtasks within any given task.

Grumman example. An example of a customer-specific application in which the interface is configured for a customer is the Grumman Aircraft Maintenance Aid (PMA), developed by Grumman. The PMA is a rugged, laptop computer designed by Grumman. It is based on a Sparc processor, but it is specially built to withstand severe operating conditions required for some military systems. The PMA runs Unix and is equipped with electronic manuals created in Interleaf 5. The PMA plugs into the data bus of a military vehicle and downloads information directly from the weapon system's computer. Maintenance technicians, interacting with the PMA through Interleaf-created documents, press on-screen buttons to view step-by-step repair instructions. Scanned-in maintenance manuals are also available and are retrieved by activating hypertext hot buttons.

In this case, the end user does not employ Interleaf's popup menus at all. The entire interface is a programmable document that contains "hot zones." Activating a zone initiates a behind-the-scenes script that brings the appropriate information to the screen.

Texas Instruments, another Interleaf 5 customer, demonstrated what it calls the Table Object Population System, an application of Interleaf 5 that fills in, or "populates," tables within data sheets with information pulled from a product database. TI expects the system to reduce by as much as 40% the time it takes to produce documentation about its semiconductor devices.

Developer's toolkit

In addition to selling end-user solutions, Interleaf is embarking on a new venture. For the first time, it is entering the programming market with a toolkit for building applications. Just as computer language vendors offer toolkits for creating, compiling and debugging applications written in that language (C, Pascal, etc.), Interleaf is selling a toolkit for building, debugging and running a publishing program. The difference is that its toolkit includes an incredibly robust engine that enables writers to edit text and graphics, build tables, and switch between them. It also allows for font scaling and rasterizing, translators—in short, everything you need to create a programmable compound document editor.

Unlike vendors that license source code to which users add extensions (typically in C), Interleaf allows extensions independent of the source code. We believe this is a significant plus for developers. It means that they can develop application-specific solutions that do not have to be rewritten, debugged and recompiled each time Interleaf updates part of its engine. Less-based extensions are executed at run-time, they need not be recompiled code. Yet these extensions can do anything from simple macros that may be represented as icons (such as Windows 3.0) to interacting with other programs (data bases, spreadsheets, scientific measuring equipment).

Various support levels. Because it wants to encourage enthusiasts, Interleaf felt it was imperative to offer the toolkit inexpensively to hackers who are not necessarily developers with lots of cash. Thus, it is offering the Methods Toolkit—the full software toolkit and documentation, but without support—for much less than the Developer's version. (As we said above, it is free to the academic market.) When they run into trouble, those who do not have a support contract will be able to buy support from Interleaf on an hourly basis.

For more formal developers or VARs, Interleaf is offering the Developer's Toolkit, with documentation, training, and support, at a higher price. Obviously, developers choosing that option will be given a higher level of support than those who try it on their own.

Third-party products

In today's market, the success of any programming language or toolkit depends on application software, particularly the third-party vendors are on hand to announce their support.

Houghton Mifflin introduced Writer's Helper, a package for Interleaf 5 that includes Houghton Mifflin's CorrectSpell spelling checker, online American Heritage dictionary and Roger's Thesaurus and CorrectText, EHS's grammar checker. The
The Publishers and, soon, Ventura). In anticipation, Interleaf left the typesetter market business a year ago, and it now relies solely on software and services for revenues. The transition has been difficult, but Interleaf has proven that it is no has-been. In software sales alone this year have matched its turnkey sales of last year.

Interleaf 5 continues Interleaf’s tradition of unbridled innovation. Believing that the value of merging text and graphics would diminish as that ability becomes commonplace in less-expensive, mass-market products, Interleaf has moved to new, uncharted territory, creating a new type of publishing application (the programmable compound document editor) and in active documents redefining what is meant by an electronic document. Without question, the publishing market is beginning to look beyond the problems of merging text and graphics to a new set of problems, such as document management, database-oriented publishing and content-based retrieval. In these areas, Interleaf is not without competition. ArborText introduced programmable objects a year ago. Daniel has been hard at work integrating Pagemaker with Oracle and IBM’s d7-tagged databases. Digital has an agreement with Verity to bring content-based retrieval to all of its products running on a network. But regardless of its fate, Interleaf’s has opened up a whole new realm of possibilities, and by popularizing the concept of active documents, will almost surely leave its mark on the industry.

Interleaf 5 is also a foremost example of extendible software that enables clever end users or system integrators to tie a publishing package with other programs. The use of Quick Xpress in the newspaper industry is the best-known example of this trend. Interleaf 5 is the first product designed specifically to address these emerging issues, and it does so in a way that goes far beyond any previous product.

What remains is for Interleaf to prove that Interleaf 5 works, and that the company can thrive in its new role of software developer own system integrator. It has won several multimillion-dollar contracts based on its technology, and installed a few at customer sites, examples of which were shown at the Conference. But it has yet to begin volume production. When it does in a few months, Interleaf will have the chance to dispel its skeptics and separate itself from its competition.

Mark Walter