Advanced Function Printing—From print to presentation

by R. K. deBry
M. W. Munger

The strength of Advanced Function Printing™ (AFP™) is due largely to the architectures that form its foundation. The architectures on which AFP is based have been developed over the last 12 years and have influenced the development of standards, competitive architectures, and, most importantly, software inside and outside IBM. Customers are demanding a more comprehensive view of printing that includes easy creation, viewing, and even specialized editing of printable documents. These "next generation" requirements are now being satisfied by software products that are based on the existing architecture. This paper describes some of these products and how they use the architecture, and describes possible future directions for AFP and related technologies.

Even though the rich printing capabilities of Advanced Function Printing* (AFP*) went far beyond the needs of most AFP customers at the time of its introduction, new requirements were soon identified that went far beyond print. The capabilities of AFP are being extended to address new requirements related to printing and other forms of presentation. This paper presents, first, printing with AFP, followed by discussions of viewing, storage, and retrieval of documents, and finally what the future holds for the AFP document.

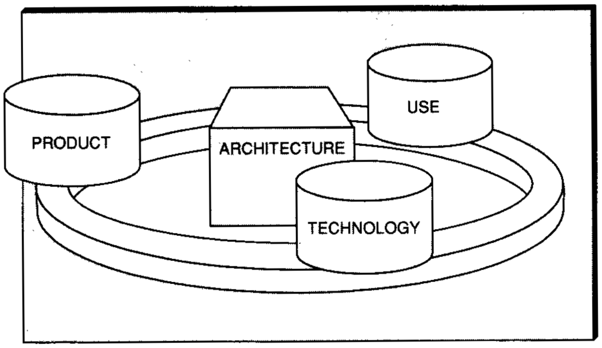
As originally conceived and developed, AFP was a system of architectures and processes aimed at printing documents on paper.[1] It is still unparalleled in system management of printers and printer resources. A printer driver interprets the presentation form of a document and adds referenced resources and printer controls to build a printer command stream in accordance with an architecture. The printer command stream contains everything the printer needs to accurately represent the user's original intentions for the content and appearance of the document and to manage the printer, even through error recovery.

Two architectures are the foundation of all AFP function. The presentation form is MO:DCA-P*, Mixed Object Document Content Architecture for Presentation,[2] and the printer command stream is IPDS*, Intelligent Printer Data Stream*.[3]

Gartner Group, in its 1990 Electronic Output Strategies Conference,[4] stated "AFP will become the *de facto* printing architecture." With the incorporation of AFP into the workstation and Operating System/400* (OS/400*) environments, thousands of AFP installations now run on every major IBM platform. Xerox supports MO:DCA-P in its Printer Access Facility, XPAF**. Low-end printers (including some driven by non-IBM data streams) are also now supported in AFP.

Figure 1 shows four areas that interact with one another. Users' needs and influences are a driving

Figure 1 Four areas of interaction



force for technology and products; technology both enables usable products and requires product improvement; products are built in response to both user requirements and new technology; and architecture is a means of centralizing all the definitions required by the products, the technologies, and the specific end-user functions. Architecture maintains order in a changing system. These four areas are used to show the dynamics of expansion of AFP into viewing and beyond.

Printing AFP documents

Use. The world of print changed dramatically and permanently the day the first laser printer was attached to a computer. The quality of computer output for the first time rivaled the typewriter and the major commercial printing processes. It is no surprise that computerized laser printing began to encroach on the domain of both the expensive, lengthy commercial printing processes and the inexpensive but unforgiving typewriter. The technology was interesting, but more interesting was the new variety of printed material emanating from computers and the resulting stress placed back on the very capabilities of the system itself.

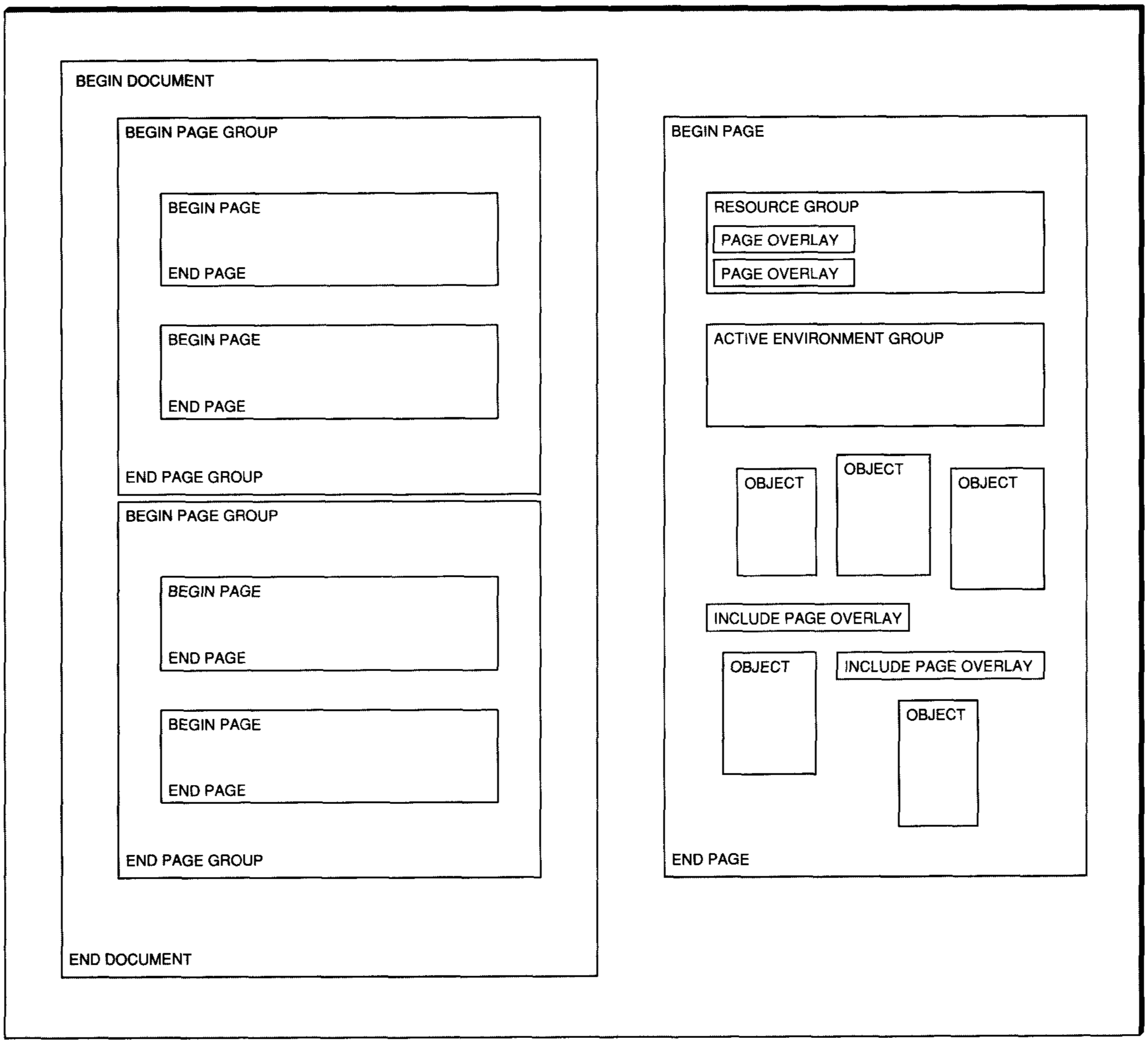
Early computer output printing was used for such things as core dumps, calculated results, program listings, and tabular data—material generally high in volume and generally not thought to be important to the corporate image. The typewriter, too, had its special purposes. It was used for typing letters, manuscripts, reports, and index cards and for filling in forms. Typewriters were (and still are) used where convenience and utilitarian quality were desired. The much higher quality of commercial printing processes is perfect for publishing and advertising, and for any material conveying image. Use of typographic fonts, graphics, and color is common. The media can be cardboard, foil, plastic, coated paper, or pulp, and as large as a billboard or as small as a business card or index tab.[5]

Every use previously made of a typewriter or commercial printing process has become a potential use for the computer output laser printer. However, many new requirements arise when laser printers replace these traditional printing methods. For example, consider the use of the printer in place of a typewriter for the printing of letters. Printing letters requires cut-sheet paper. Cut-sheet paper handling leads to consideration of multiple input and output bins. Cut-sheet paper handling is also more susceptible to jamming than continuous forms, raising the subject of error recovery. Can company letterhead be printed on demand, rather than requiring physical letterhead to be loaded into the printer when needed? If handling company letterhead is feasible, the need for color may arise. Availability of typographic fonts is a must. A post-processor to fold and stuff envelopes may computerize a manual step, but only if the envelopes are addressed. Addresses often include a postal bar code with its tolerance requirements. In other words, one "simple" new use of the laser printer cascades into a profusion of development items for the printers, the printer drivers, pre- and post-processing equipment, and application programs. New applications in publishing and advertising, for example, present far greater and more complex challenges to computer output laser printing.

How can large numbers of people separated by company, country, and national language cooperate to accomplish such complex goals as those described above? The key to worldwide cooperation in computer development is rigorous, well-defined architecture. Literally every function must be expressed in open, extendable, controlled interface architecture so that growth is nondisruptive and yet unconstrained by proprietary mysteries.

For high-quality, production printing, the architecture must be capable of expressing all of the functional and data elements of a document in a way that does not depend on any characteristic of the eventual presentation device. The importance

Figure 2 MO:DCA-P documents, page groups, and pages



of this attribute will become more evident in the following paragraphs.

Architecture. The Advanced Function Printing Data Stream (AFPDS) was described by deBry et al.[6] as one of the architectures of AFP. AFPDS began as a product data stream in the AFP environment. AFPDS has evolved over time into the more robust MO:DCA-P, which describes an electronic document. A complete description of the MO:DCA-P architecture can be found in the published reference manual; some of its important elements are summarized in the following paragraphs.

Just as pages are the substance of a physical document, pages are the substance of the MO:DCA-P document. Figure 2 depicts a document and its pages with their component parts. Pages may be grouped to provide a single reference for one or more pages. If present in the page, the resource group is positioned first and is followed by an active environment group and any combination of objects and "include page overlays."

The resource group contains only page overlays, which are identical to pages except that they do not contain a resource group or an include page overlay. The active environment group contains the controls required to correctly render the information in the page image, such as a list of fonts referenced in the page. Following the active environment group is the information content of the document which is carried in presentation text objects,[7] image objects,[8] graphic objects,[9] and bar code objects.[10] The objects contain data and an object environment group to specify parameters local to the object area.[11] The include page overlay names a page overlay and positions it within the current page. The page overlay is used to present multiple data objects in the same area, e.g., a form and its data.

The active environment group and objects contain further levels of detail not important here. However, the concept of structure is important. Rather than being a single sequence of function and data, the structure is blocked, having explicitly delimited elements. Blocking is the means of limiting the scope of parameter effects.

Many documents rely on reusable parts, such as fonts[12] and overlays. The reusable parts, or "resources," as distinguished from "resource groups," are created once and stored in "resource libraries" for reference by applications and device drivers. Whether a MO:DCA-P document is self-contained or references resources in a library, it is still portable to other systems. Resources, being block-structured and self-describing, are also individually portable.[13]

The block structure of MO:DCA-P is not only convenient for document construction and portability, it is ideal for various tasks of document management and processing. For example, to ensure future faithful interpretation of an electronic document, resources may be stored with the document. In contrast, to minimize storage requirements, the document may be stored with references to resources instead of storing the resources themselves.

Product. Figure 3 shows a simple model of the AFP environment. A good example of an application is the Document Composition Facility (DCF*). DCF converts text and controls entered by the user into a fully formatted MO:DCA-P document and then stores the document on the system spool to await print. Fonts, page overlays, and other resources referenced by the document are previously installed in resource libraries for reference during formatting and printing.

Blocking is the means of limiting the scope of parameter effects.

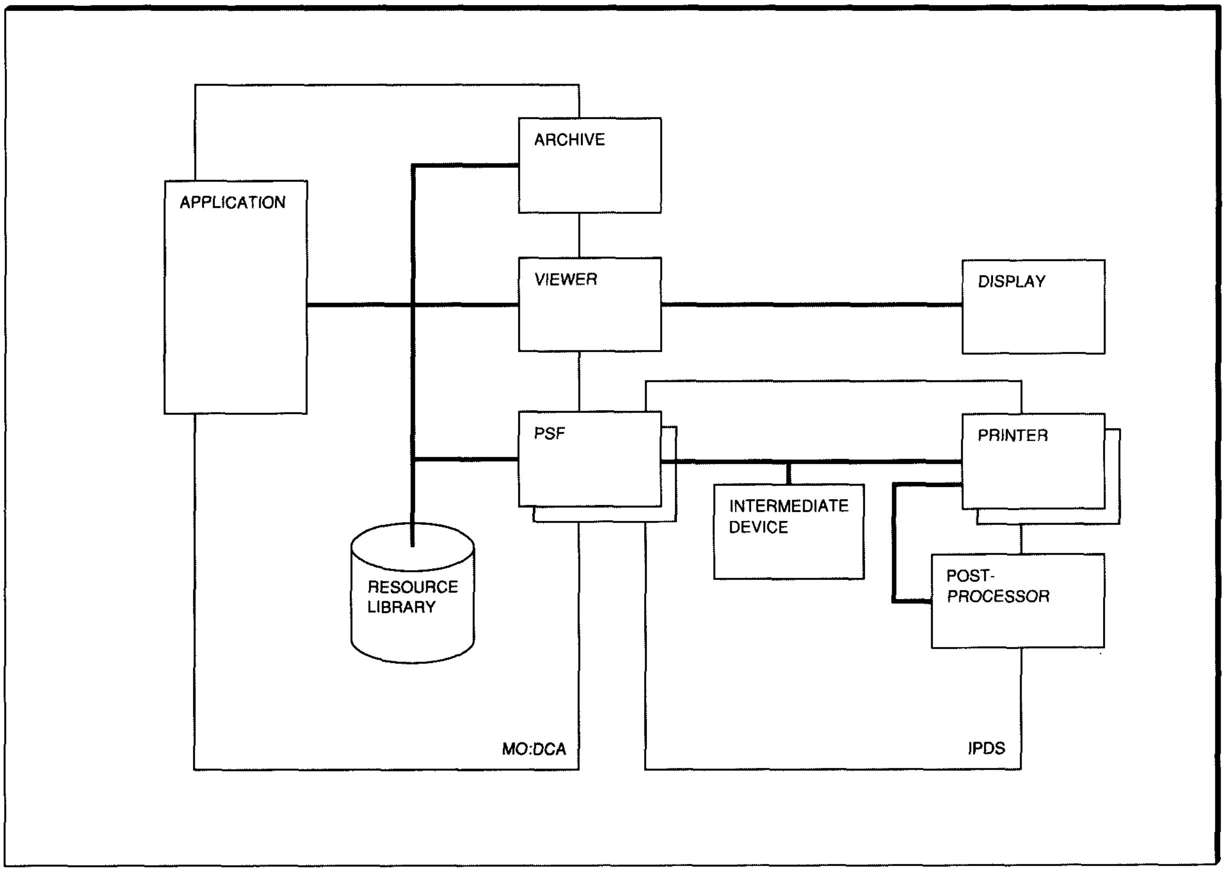
Print Services Facility (PSF*) is IBM's AFP printer driver. PSF translates the MO:DCA-P document into a stream of printer commands, IPDS. Fonts are located and downloaded to the printer, if necessary. Other resources are obtained and included in the printer command stream as required. The translation provided by PSF is relatively straightforward: the data content of the MO:DCA-P document is passed through unchanged, and the structure and parameters are translated to the similar structure of IPDS.

Viewing AFP documents

Technology. Technology costs continue to decline. Several technologies are contributing to extending AFP: in particular, outline fonts, more powerful processors, higher-resolution displays, faster display adapters, and graphical programming environments, such as Operating System/2* (OS/2*) and Microsoft Windows**.

Raster fonts are being replaced by outline fonts. Now, whether occurring on the host, on the workstation, or in the printer, rasterization can be done at nearly any size, resolution, and orientation, and on demand. Libraries are vastly smaller, one-fifth the size of raster-font libraries. Also, along with outline fonts come relative metrics, removing the barriers imposed by fixed metrics. Processor speeds in the workstation (at a given price) are 25 times faster than they were 10 years ago. Increased power enables compute-intensive pro-

Figure 3 AFP system model

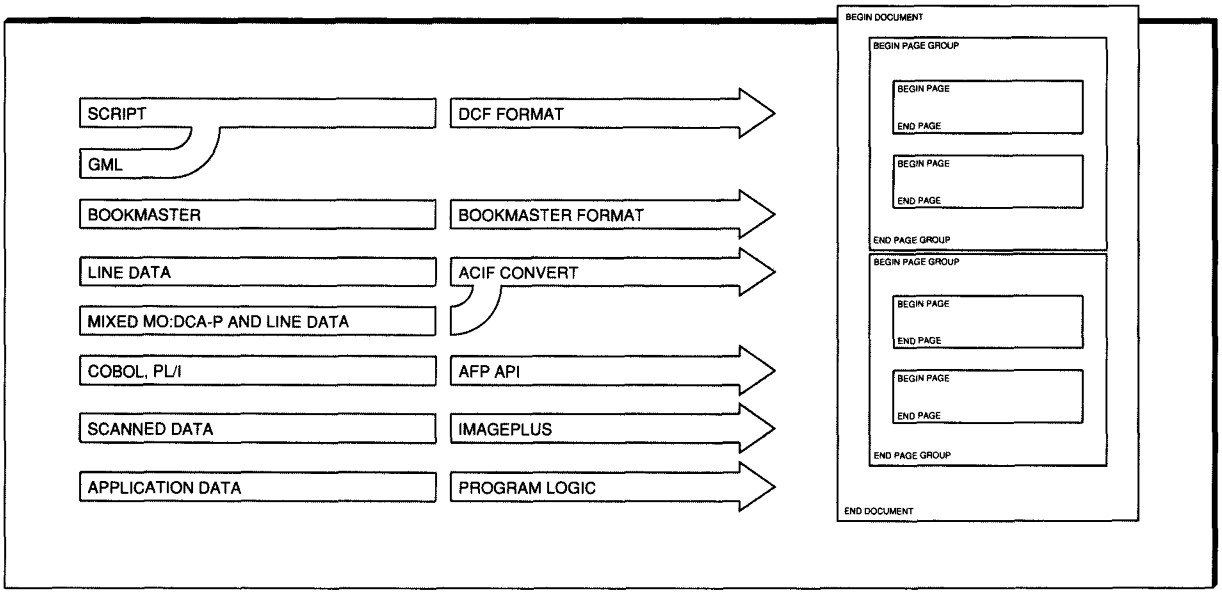


cessing such as font rasterization to be done on relatively inexpensive workstations. Vastly improved displays and the graphical user interface have made it possible and affordable to view MO:DCA-P documents with acceptable fidelity. These things taken together have made the workstation a practical document-viewing device.

Use. What is required for viewing? There must be an effective means of faithfully imaging a page and navigating to the page of interest.

Imaging requires that the workstation display the pages of the document exactly (within reasonable latitudes) as they would appear if printed. This means that all the resources available at print time must be available at display time. Considering mainframe formatting and printing as the basis, workstation viewing requires transport or duplication of mainframe data and processes, such as fonts, page segments, page definitions, form definitions, and overlays, as well as the documents themselves. Users typically have many large documents. For a complete solution, they need help in managing the large volumes of data across the two environments, or they need a facility for direct viewing of mainframe documents without first moving them to the workstation. Fonts, for example, must be available in the same typefaces and with the same attributes as those used when the document was formatted. And text is not the only data type: graphics drawing orders require equivalent interpretation on the workstation, and image data require decompression and resolution correction. Imaging requires rendering of the entire document to match the printer's rendering as closely as possible.

Figure 4 Sources of a MO:DCA-P document



Further, one rendering per page is not adequate. The user might want to enlarge or reduce the size of the displayed page, depending on the user's viewing purpose. Imaging for display must be fast, as well as faithful to the printed document.

Navigating requires being able to control forward and backward movement and to specify and find a target for viewing within the document. As seen in Figure 4, a MO:DCA-P document may come from several different sources. In any case it may be viewed with scrolling and keyword searches, but tags are required to have full-functioned navigation employing named targets. Tags are special constructs in the document that identify a page or page group as special; they carry descriptive information that pertains to that page or page group.

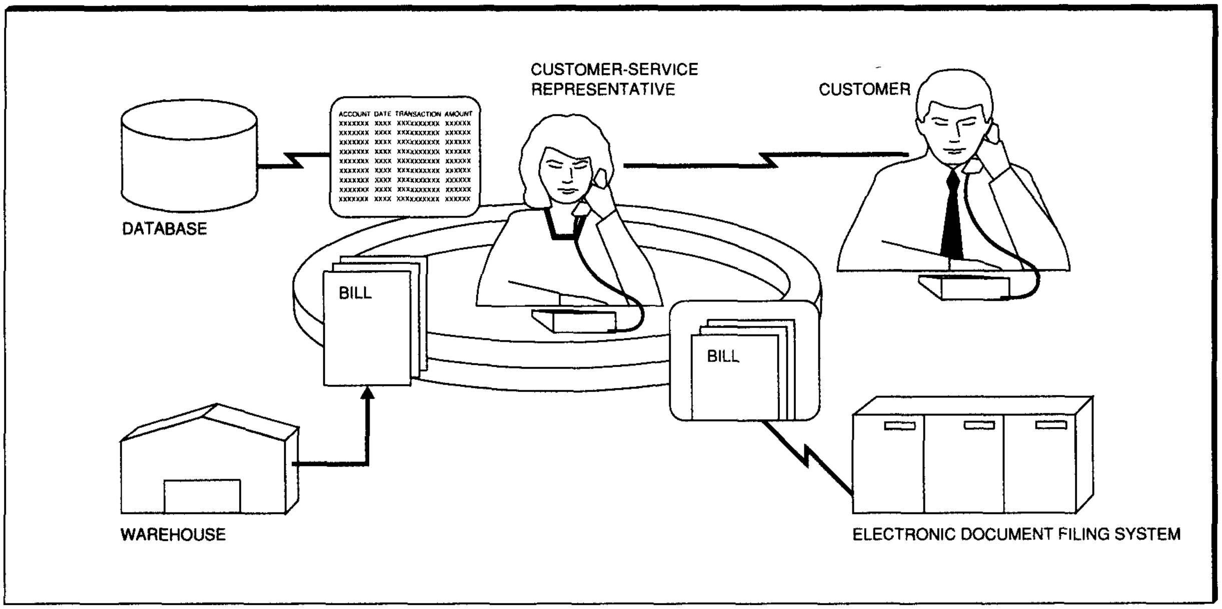
For example, in a document of telephone bills for many customers, one tag may contain the customer's name and another tag the customer's telephone number. Viewing now can be guided by a search for a specific telephone number or a customer's name, rather than a manual or character string search. (The collection of all tags in a document extracted for processing efficiency would serve as a simple index. Indexing is an emerging technology, about which more will be said later.)

The stored document becomes a type of database, with the additional benefits of being immediately printable and looking the same to the viewer as it does to the reader of the printed version: a customer-service advantage.

Let us continue with the customer-service example as depicted in Figure 5. Suppose a telephone customer calls with a billing question: "The billed amount seems too high; the bill is five pages long and many phone calls do not appear to belong to me. The monthly rate also seems to be too high." The customer-service representative will respond, probably using one of three common approaches to customer-service support: database, physical files, or the electronic document.

The database approach is satisfactory. Keying on telephone number and month, the customer-service representative has all relevant data immediately available. "But all the calls on page two do not belong to me," is a statement that can only be discussed if the customer-service representative has knowledge of page content. Without pages, a prolonged discussion will follow regarding the individual calls on "page two." Not having the concept of page is a drawback to using the database approach.

Figure 5 Customer-service example



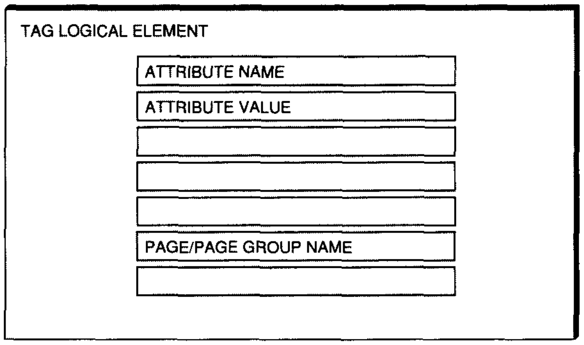
The physical files approach is also satisfactory. All of the customer's questions have a complete reference base. But physical documents can take days to retrieve, not only delaying resolution but also requiring refiling. The statement, "But if I look at December's bill the rate was lower," can only be discussed after the customer-service representative has also retrieved the bill for December (and all bills for this year and last, just in case). Refiling is an additional expense.

Electronic documents offer a solution to the shortcomings of both methods. Retrieval (and refiling!) time is measured in milliseconds, and the retrieved information looks the same to the customer-service representative as it does to the customer.

Though viewing has become technically feasible, the existing environment must be preserved as viewing is introduced. Architecture satisfies transition and compatibility needs and, at the same time, provides a well-defined basis for change.

Architecture. MO:DCA-P is the basis for all AFP viewing and printing. MO:DCA-P lends itself well to the construction of page images for the display

Figure 6 Tag logical element



that are faithful to the printed rendition. Viewing, however, requires two primary architecture changes: tagging of key material within the document and document indexing.

Locating key items of information in a document involves identifying and tagging the items before viewing to permit efficient and structured searches. Tags have been added to the MO:DCA-P architecture to be inserted into the document either during or following formatting. Figure 6

Figure 7 Tag logical elements in a MO:DCA-P document

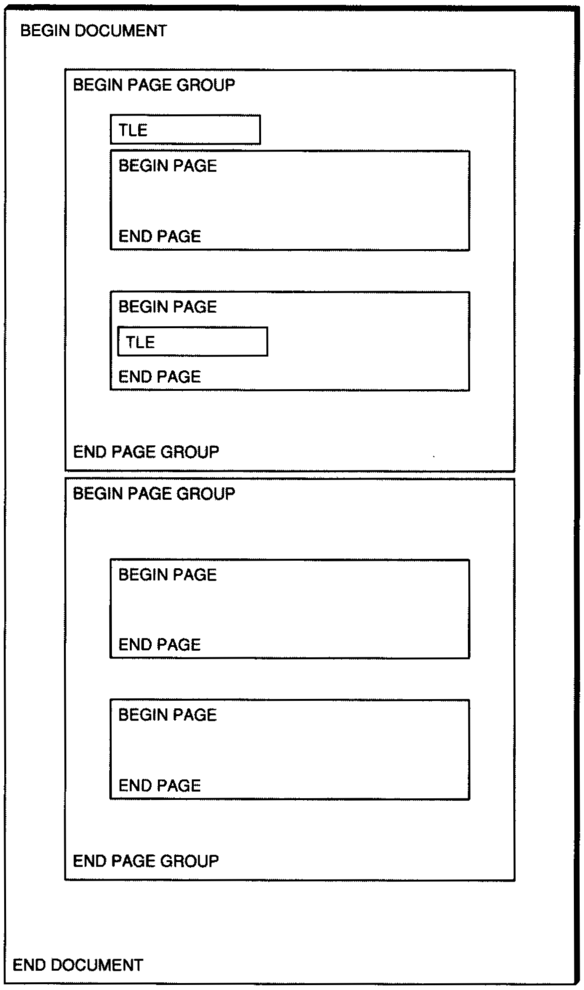


Figure 8 Index element

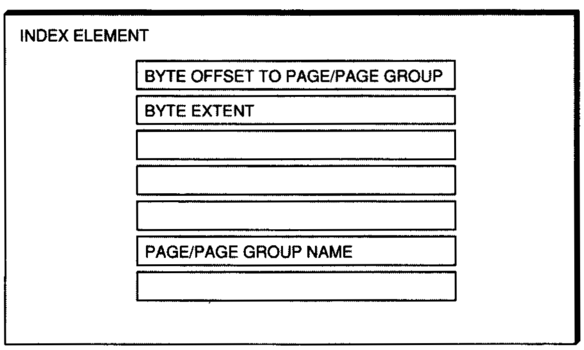
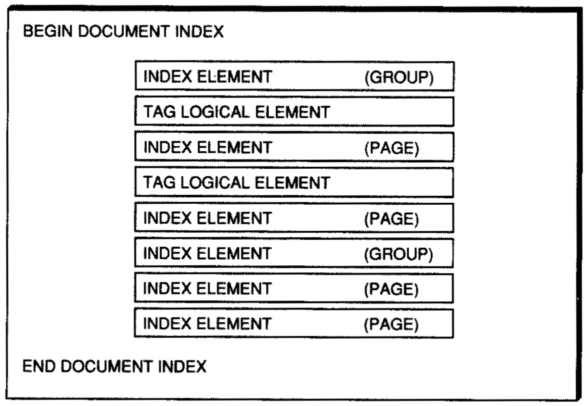


Figure 9 Document index



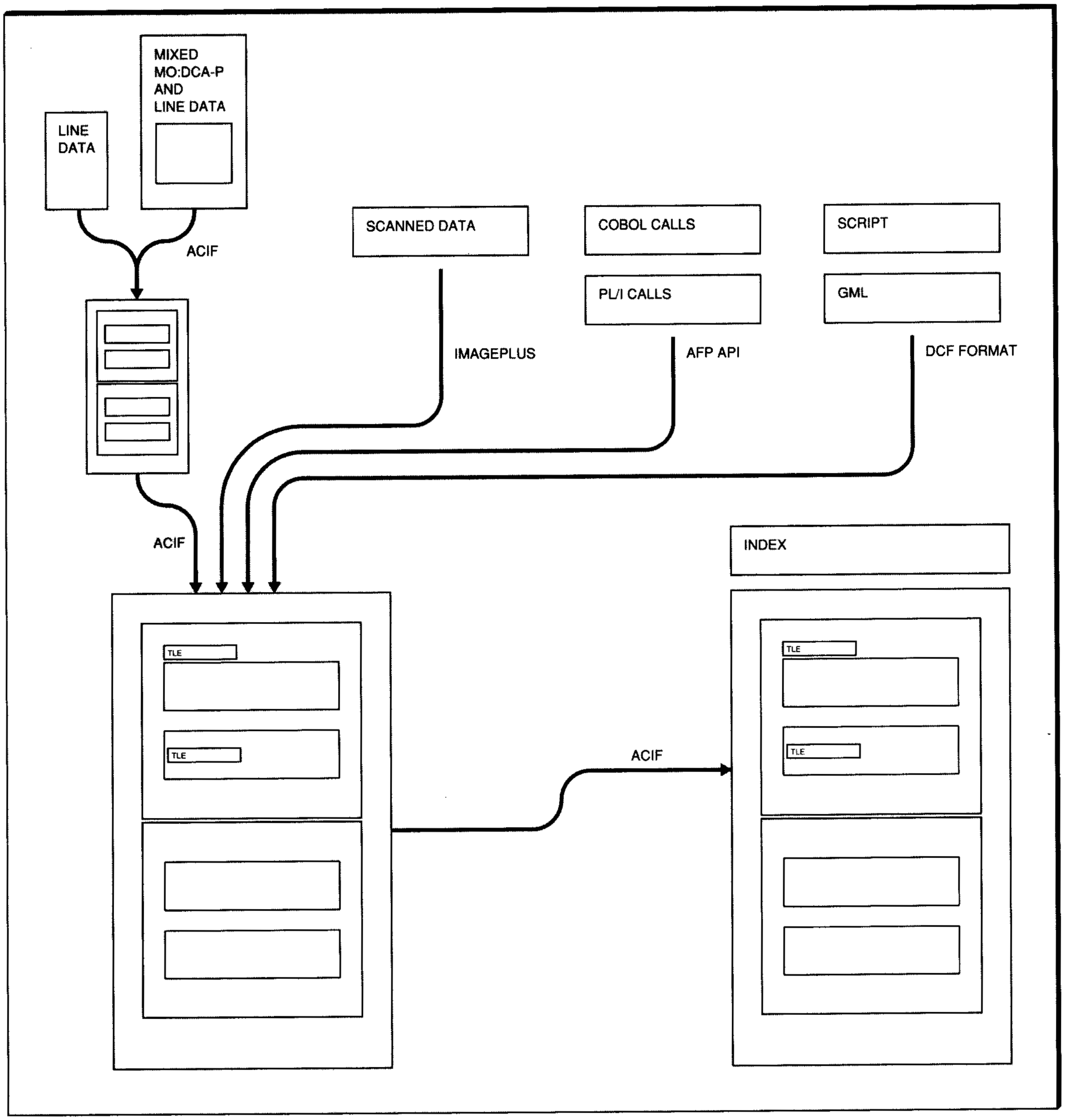
shows the structure and the main content of such a tag, called a tag logical element (TLE). The key parameters are *attribute name* and *attribute value*. Page name and page group name are not allowed in the TLE when placed in a document, but may be required for TLEs in a document index to provide the association with an index element (see the following paragraphs). Figure 7 shows where the TLEs may be found in a document. The TLEs may be placed first in a page group or anywhere in a page where the major components of the page may be placed, but not within any major component (e.g., image object or resource group).

An index comprises a sequence of TLEs and index elements (IEL). The index elements, as seen in Figure 8, contain calculated offsets to and extents of each referenced page or page group. Figure 9 shows the structure of a MO:DCA-P document index.

The document index provides a displacement into the document for the selected key subject. Each TLE relates, or maps, a subject to page name or page group name. Each IEL maps a page name or page group name to a displacement and extent. There may be one IEL in the index for each page and page group in the document.

The tag and document index are the two major architecture updates that have made viewing possible. These updates are the basis for two new AFP

Figure 10 Sources of a tagged MO:DCA-P document



products that work together to faithfully display MO:DCA-P documents.

Product. Figure 10 depicts some possible sources of the tagged MO:DCA-P document. AFP Conversion and Indexing Facility (ACIF*) is an MVS- (Multiple Virtual System) and VM-based utility that provides three major functions: conversion of line-mode data (or line-mode data mixed with MO:DCA-P) to pure MO:DCA-P, a facility for intro-

Table 1 ACIF document indexing

Insert TLEs= Y Index pages= Y	Insert TLEs= N Index pages= Y	Insert TLEs= Y Index pages= N	Insert TLEs= N Index pages= N
Begin document index	Begin document index	Begin document index	Begin document index
IEL—group	IEL—group	IEL—group	IEL—group
TLE	TLE	TLE	TLE
TLE—new		TLE—new	
IEL—page	IEL—page		
IEL—page	IEL—page		
TLE	TLE		
IEL—group	IEL—group	IEL—group	IEL—group
TLE—new		TLE—new	
IEL—page	IEL—page		
IEL—page	IEL—page		
End document index	End document index	End document index	End document index

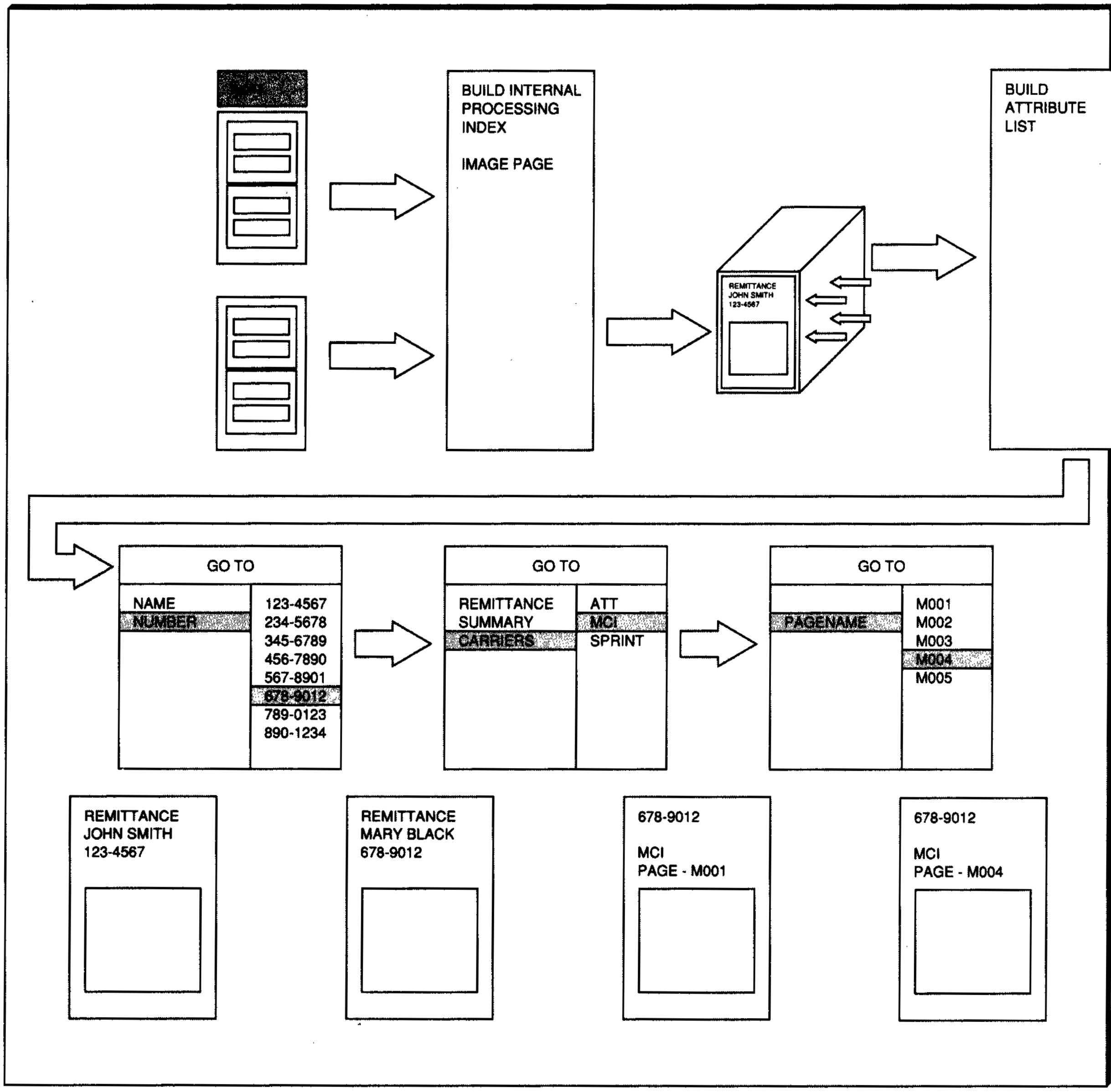
ducing tags, and document-index building. Line-mode data originate in applications written for line printers, and these data represent the majority of data printed in the AFP environment. Also, for a pre-existing pure MO:DCA-P document not prepared for full-functioned navigation, ACIF provides an exit as a means for the user to introduce tags. The user is guided through the necessary steps to introduce all tags required for his or her viewing needs. A new MO:DCA-P document, including tags, may have been created using the AFP application programming interface (API),[14] DCF, or ImagePlus*. Telephone bills, for example, are created by an application accessing various tariff tables, customer files, and control information to custom build each customer's bill for the month as pages in a MO:DCA-P document. The AFP API not only automates much of the building of the MO:DCA-P document but also provides a means to insert TLEs as each page is built.

Once the tagging of the MO:DCA-P document is complete, a document index can be built. The document index is also built on the host using ACIF. Table 1 shows the indexes that could result from the processing of the document shown at the bottom of Figure 10. Two parameters control the TLE content of the output document: one causes TLEs to be inserted or not inserted; the other causes the index to include both page-group-level and page-level TLEs or just page-group-level TLEs.

The document index may be internal or it may be external to the document, so the user or using application may make the trade-off between rebuilding of the document index when needed (such as for a "small" document) and building the document index once and storing it away for future use (such as for a "large" document). The document index is built by searching for tags, copying and creating TLEs, and creating IELs with their respective page and page group displacements, and placing them in order in the MO:DCA-P document index just described. The document index then enables efficient navigation based on information that is pertinent to the user.

The AFP Workbench* is an application based on Microsoft Windows. Figure 11 shows an overview of AFP Workbench viewing operations. Once AFP Workbench is active, a document may be selected and opened. (This discussion assumes that documents to be viewed have already been downloaded.)

Figure 11 Workbench overview viewing



To display a page that is faithful to the printed rendition, AFP Workbench accepts MO:DCA-P documents that include imbedded overlays and page segments, compressed and uncompressed image data, and form definitions, or formdefs. AFP Workbench permits the user to export overlays and to select and extract specific pages for printing or other uses. AFP Workbench uses Adobe Type 1 fonts; because of the wide variation in font capabilities of existing printers and displays, exact pixel alignment is not possible in all cases.

When a document is selected for viewing, an internal processing index is built by either searching for and using all the tags in the document or by translation of the document index. If the document index exists when AFP Workbench accesses the document, the building of the internal processing index for AFP Workbench is faster than if there is no document index. The first page of the document is imaged for the user to begin viewing. With use of the internal processing index, an attribute list is built in anticipation of a request to go to another location in the document. To find a particular location in the document, the user selects GO TO and selects the subject (i.e., attribute) of choice, e.g., telephone number. The AFP Workbench then presents a list of telephone numbers (i.e., attribute values). When the user selects a telephone number, the location of the page containing that tag is picked up from the index and the page is retrieved, then rendered for the specific display, and presented to the user. If the page rendered is the first of a set, the user has another opportunity to select a subject, such as carriers (telephone companies), and then to specify which carrier. The user can continue in the same manner until the desired page is presented. At any point in the navigation, the user can page forward or backward as an alternative to selecting named targets.

The AFP Workbench permits navigation with or without TLEs. Without TLEs, navigation is limited to scrolling, relative page numbers, and keyword search.

In summary, ACIF provides a host-based "conditioning" of MO:DCA-P documents: converting to proper format, introducing tags, and building an index. AFP Workbench provides a basic view capability, with keyed retrieval of pages (and page groups), and a variety of import and export functions. Taken together, the two programs provide a flexible host-workstation solution to the AFP user.

The ability to view a document requires controls related to the flow of work in the establishment ("workflow"), much like those used for the printed document.

Use. The availability of viewing functions suggests paper avoidance in situations normally demanding paper as the medium of communication. For example, legal, business, and technical applications typically involve not mere document "creation," but document "development." A document may develop over a year or more and be reviewed by 100 or more readers. With appropriate end-user functions, viewing could ease the entire review process and remove the expense of managing physical documents. Two additional

ACIF provides a host-based "conditioning" of MO:DCA-P documents.

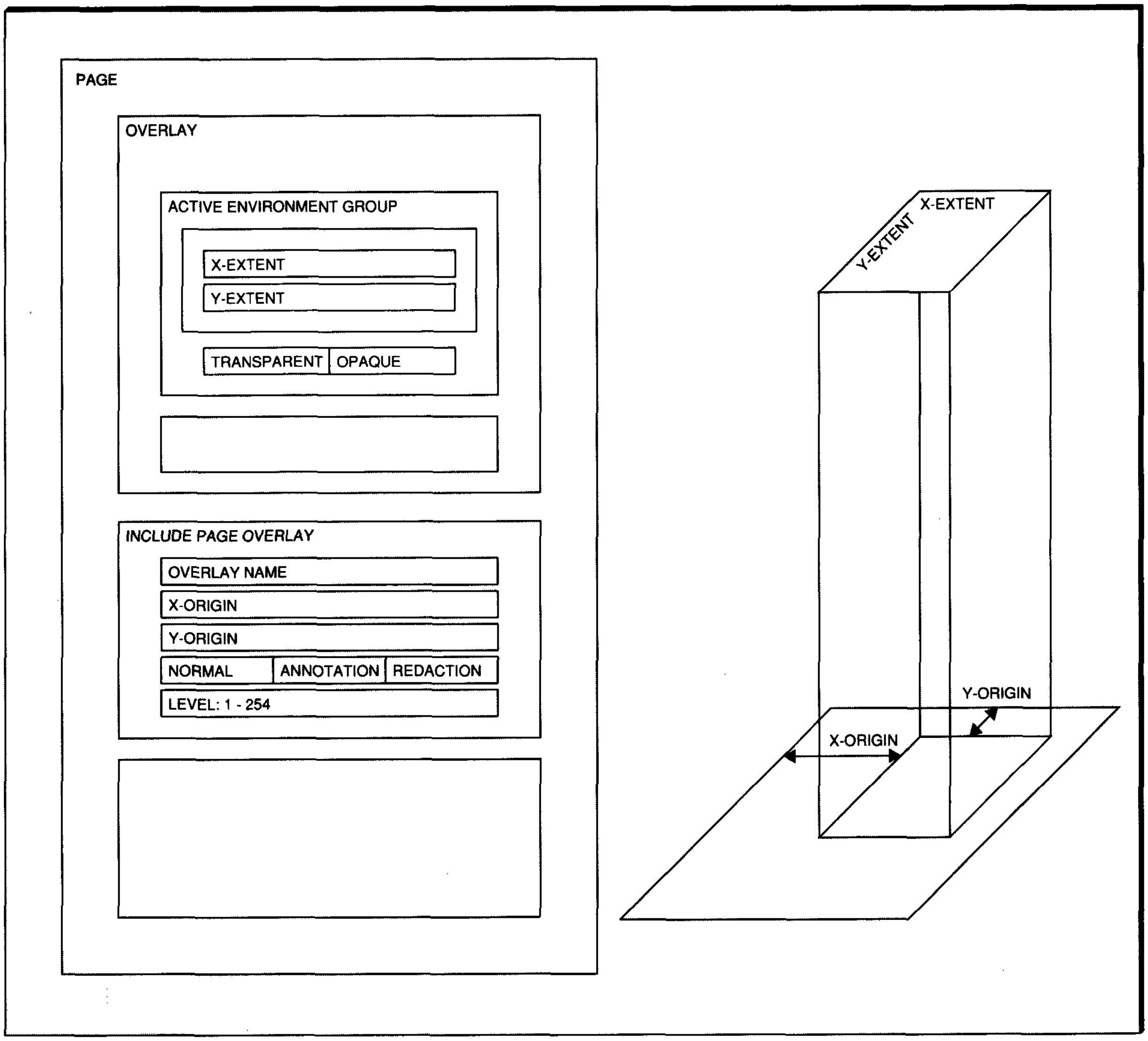
functions add value to the viewing capability: the ability for each reader to make notes on the viewed document (annotation), and the ability for the author to control *what* readers see *which* parts of the document (redaction).

Annotating a document with material that is related to, but not part of, the document requires the capability of creating the annotation, associating it with a point in the document, and selecting it for optional viewing or printing.

Remembering that a MO:DCA-P document can contain a wide variety of data types, one would also expect those same data types to be valid as an annotation. Attaching an audio object to a text page has obvious application in telephone-based customer-service operations. Attaching a bar code object to an image or graphics object permits mechanical identification of scanned photographs and engineering drawings. Annotation should permit information expressed in any MO:DCA-P data type to be associated with any point in a MO:DCA-P document. When the page is presented (i.e., viewed or printed), a visible marker should denote the presence and location of each annotation. Time and date should be a part of each annotation. In whatever way the annotation is implemented, it cannot corrupt the document or its data. The annotation must be separate from, but associated with, the data of the document.

"Redact" means "to select or adapt for publication."[15] In the context of electronic document processing, one can envision redaction as a dynamically applied suppression. An ability to define user levels and corresponding document

Figure 12 Include page overlay



information levels is needed. Each presentation of the document can then be tailored to the level of the user. Information, whether document or annotation, of a higher level than that of the user is then not presented. With annotation and redaction, document viewing can be an aid to many commercial and scientific workflow applications.

Architecture. A test of the durability of an architecture is its adaptability to new requirements. What at first appears to be a complex function is satisfied by a simple addition to MO:DCA-P. Both annotation and redaction are accomplished in MO:DCA-P via two conditional processing parameters that have been added to the include page overlay, as seen in Figure 12.

The page descriptor in the page overlay both specifies the X and Y dimensions of the image of the overlay and indicates whether the overlay should be transparent or opaque when applied to the page image.

Table 2 Overlay logic

Include Type	Level Include::User	Processed?	Overlay
Annotation	<=	Yes	Transparent or opaque
Annotation	>	No	–
Redaction	=>	Yes	Opaque
Redaction	<	No	–

As described above, the include page overlay specifies within the page image where to place the origin of the page overlay. Added to the include page overlay was a pair of conditional processing parameters that determine whether to process the include. One parameter indicates whether the include is an annotation or a redaction; the other parameter specifies the levels at which to process the include.

The objective is to include an annotation for users of *adequate* level and include a redaction for users (applications) of *inadequate* level. Table 2 outlines the overlay logic. As an example, consider a document that contains annotations and redactions of three levels each: one, two, and three. An application, which has been assigned a level of two, selects the document for viewing. Annotations of level one and two will be included, and redactions of level three will be included. The viewed pages will be displayed *with* all the information in the level-one and -two annotations and *without* the information obscured by the level-three redactions.

Page overlays, like pages, may contain any valid data object, which satisfies the need to annotate various data types with various data types. Applications are consequently able to incorporate viewing as a document development and management tool to save both time and paper. However, expanded uses for electronic documents tend to increase their number: demands on both storage and system input and output volumes increase commensurately.

Storage and retrieval

Technology. Technology is responding to the need for systems to accept, store, and produce an ever-increasing volume of documents. Cost of storage on a mainframe computer has decreased by a factor of ten over the last ten years[16] and on the workstation by a factor of four.[17]

Drastically improved economics of storage encourages expanded usage. With bar codes, optical character recognition (OCR), magnetic ink character recognition (MICR), and improved scanners and recognition algorithms, capturing, storing, and retrieving *data* have become much easier and more efficient. Unfortunately, we do not have easy and effective methods for identifying and retrieving *information*. The following paragraphs identify the distinction between data and information, and the problems encountered in regarding data as information.

Use. The distinctions between data and information are important. Good customer service, for example, requires easy access to information, not simply to data. That information should be relevant to each customer, especially to provide real-time responses to telephone inquiries. This requires that the information be in usable form. Large service companies have literally millions of documents on file and, though electronic storage and retrieval may be fast, effectiveness depends on the capability to retrieve and manage information.

Information is appropriate data arranged in usable form, such as a document. "Appropriate" and "usable" are subjective, even abstract, words that expand the definition of information far beyond the capabilities of database, and even beyond artificial intelligence. To be sure, a database does offer information, but only in a limited sense. Retrieval of data from a database is very straightforward, but discovery of rationale and new relationships and the expression of those ideas in readable prose requires additional effort. Documents, in contrast, already express ideas of relationship and rationale in readable prose, but selection and retrieval require additional effort. As more documents are stored in electronic form, the difficulty of effective retrieval heightens and becomes more obvious. Information science has made good progress in the indexing and retrieval of entire documents, but progress toward indexing and retrieval of appropriate excerpts of documents has been slow. Indexing is still manual for the most part and is accompanied by some degree of subjectivity.[18] Electronic documents are indexed as they are created or stored. Indexes on forms can be created economically if keywords can be used. Indexes on images require some manual preparation and may employ OCR, MICR, or bar codes.

Indexing in general, however, is complicated and costly. To explain, let us consider two simple methods and note their shortcomings. One method would be not to index at all, which results in searching entire libraries for keywords and phrases. Another method would be to mechanically index every word, as in a concordance. Other than the fact that the methods apply only to text documents (and that electronic documents may be scanned in as image only), these approaches have some serious problems: performance, vocabulary, and precision, to name three. The searcher may not understand the vocabulary of the subject being researched and may not be able to formulate meaningful queries. The searcher may know the vocabulary but not be able to focus the search tightly enough to select a few good references rather than hundreds of general references. A usable index needs to have as standard a vocabulary as possible (which may not match the vocabulary of the reference document) and be a knowledgeable extraction of the relevant from the irrelevant (which is subjective). A good index today may not be a good index five years from now. In spite of these difficulties, the index cannot be avoided: the usefulness of documents as information depend on the index. The rigorous, useful index on a library of full-text documents is a major challenge. Though much work has been done on information indexing, it is still regarded as an emerging technology,[18] especially for nontextual data.[19] An excellent case study is presented by Plesums and Bartels.[20]

Nonetheless, users of document-imaging systems recognize that effective storage, retrieval, and viewing functions can be developed and are significant to state-of-the-art customer service.

Product. Whereas AFP Workbench has taken a definitive step toward *document* content indexing, ImagePlus features *library* content indexing. These two products are establishing an important technology basis for effective information retrieval.

ImagePlus is a family of document-imaging systems (in several operating-system environments) that provide for the capture, indexing, storage, viewing, printing, and distribution of large quantities of documents. The product family incorporates high-speed document scanning, workflow management, and storage capabilities.

ImagePlus/2 (the OS/2 implementation), for example, allows the user to electronically enter a document into the system by a scanner or facsimile, index and update the document, and view the document, both on entry to validate quality and for subsequent reference. The document may be assigned to a class and put in a folder or workbasket for later retrieval and processing. Documents can also be sent to a printer or sent by facsimile transmission.

Work is always in progress to improve interoperability. One notable area already achieved is the AFP printing and viewing of ImagePlus documents using MODCA(P) Interchange Set 1.

ImagePlus implementations provide simple, but powerful, annotation function. ImagePlus documents are stored in MO:DCA-P form: the MO:DCA-P overlay contains the annotation. The position of the overlay in the document and the position of the information in the overlay determine the association of the annotation with a point in the document. ImagePlus provides a means to suppress or include the overlays when the document is printed or viewed.

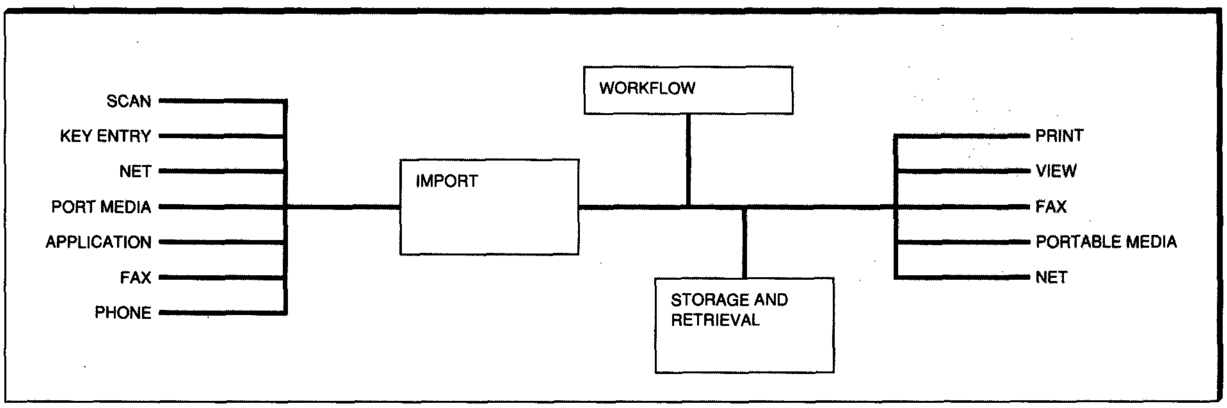
Viewing functions include pan, rotate, invert, and zoom. Capture and view include bi-level, color, and gray-scale images. Pages consisting of both image and text data can be displayed or they can be printed on ASCII and AFP printers. ImagePlus is an excellent example of how technology and architecture can be put to use to serve customers more efficiently.

Printed documents have been used in innumerable ways. Electronic documents were once only used for printing. Now customers are attempting to use electronic documents in the ways in which they formerly used the printed document, such as group reviews during document development and customer-service reference. Technology and architecture are responding to the expanded view of the electronic document.

Future outlook

Technology. Multimedia products now demonstrate time-synchronization of audio with video

Figure 13 Document-imaging system model



or graphics. The many forms of audio and video make it easier to integrate them with other forms of data. Color print is now affordable and of presentation quality. Font technology is continuing its evolution to fit into the workstation environment.

Hypertext is growing as the power of workstations grows. So far, the major hypertext applications are not based on a printable document. Merging hyper function with print function in a single form that could be both printed and navigated in hypertext fashion would improve overall usability and reduce storage requirements.

Printing will soon be expected to provide full process color with all data forms. Viewing will soon be expected to provide audio and video, in addition to those functions found in print.

Tasks that are cumbersome with a printed version of a document are good candidates for electronic methods. Examples are document development, document review and comment, and cross-referencing. If the electronic method is weak, however, so will the efficiency and resulting customer service be weak. For example, if hypertext is implemented for phone bills, but only allows looking at on-line (i.e., this year's) bills, customer service suffers. Strength is required. The solutions must be robust, durable, and flexible. Programming can do much to deliver function, but programs age, become inadequate, and are replaced. The most important ingredient for robust, durable, and flexible electronic methods is strong architecture. AFP is based on the Mixed Object Document Content Architecture, which provides the needed strengths for expansion.

In the following examples, note how simple the architecture changes seem in comparison with the intended uses and vast implications to computer users.

Architecture. Use of color engenders some complexity. The simple observation that both transmissive and reflective colors are required for display and print, respectively, demonstrates the need for multiple color encodings and transforms from one medium to another. Today, color tables are defined in MO:DCA-P. A color table relates the components of a standard color model, such as "red, green, blue," to control codes for use in specifying color for MO:DCA-P objects. As additional needs for color arise, tables for each color model can be added or expanded with minimal effort.

The hyper-document is implemented in a number of different ways; most are not printable. Today MO:DCA-P is being enhanced with a robust link that begins the journey toward full hyper-document function. Perhaps the link will play a role in the database question mentioned earlier.

Time-sequencing is a dimension only recently associated with documents. As multimedia grows and takes shape as a tool and a technology, the function of time-sequencing is bound to appear in documents. Time-sequencing and other multimedia functions are ideal for workstation-based presentations, making use of audio and both still and motion video. MO:DCA-P is well-structured to receive such modifications.

Even with enhancements in these areas, document architectures will be stressed by users' applications.

Use. The printing of a document and the use made of a printed document are independent. However, the electronic document and its use are tightly interwoven. The usage of such products as AFP Workbench, ImagePlus, and all the document imaging systems defines new requirements for the electronic form of the document. Figure 13 diagrams a document-imaging system with a sampling of functions. The document-imaging systems listed in Table 3 implement MO:DCA-P, either directly or through import.

Document-imaging systems are an excellent bellwether of the document environment for two reasons. They are a natural solution for any business needing records management, archival facilities, or customer service,[18] a wide-ranging class, indeed! Second, they are a general case of many different simple document processing configurations. The workflows and work products represented in these systems may be very different, stretching the electronic-systems capabilities in various directions to meet business needs. The stretching is the force that will cause the introduction of new technologies. As users lead products and technology into the future, it will be the adaptive that survive and the responsive that excel.

Summary

The history of the electronic document has been traced from an AFP perspective. Viewing, in particular, has been described to demonstrate how products and architecture are responding to increased demand for function. Specific examples in AFP Workbench, ImagePlus, and MO:DCA-P were used to illustrate expansion into new areas beyond print. And finally, prospects for the future of the electronic document were offered, showing the electronic document as the key to effective workflow in the establishment.

Table 3 Document-imaging systems implementing MO:DCA-P

Alpharel, Inc.	DDTMS
Computron Technologies Corp.	Epic/Workflow
Hewlett-Packard Co.	Advanced Image Management System
IBM	ImagePlus
Image Business Systems Corp.	IBS Image System
SAIC	MOSAIC
Scan-Graphics, Inc.	Friends, FriendsPlus
Sigma Imaging Systems	OmniDesk

Acknowledgments

The authors would like to acknowledge the developers of the many products referenced and described in this paper, particularly the AFP Workbench, which represents a significant step in the expansion of AFP from the print-only mainframe environment. J. R. Sampson has been the key technical leader in the birth and development of the Workbench. We would also like to thank the architects for helping us maintain an orderly growth in our data streams, especially R. H. Hohensee, who is the current owner and developer of MO:DCA. And we thank J. K. Brand for his excellent help in matters of grammar and style.

Cited references and notes

1. R. K. deBry and B. G. Platte, "Advanced Function Printing: A Tutorial," *IBM Systems Journal* **27**, No. 2, 219–233 (1988).
2. *Mixed Object Document Content Architecture Reference*, SC31-6802, IBM Corporation; available through IBM branch offices.
3. *Intelligent Printer Data Stream Reference*, S544-3417, IBM Corporation; available through IBM branch offices.
4. R. Sherry, *The Five-Year Planning Scenario for Electronic Output Strategies*, Gartner Group, Inc. (August 1990), p. 5.
5. . . . or smaller! Monotype Corporation was giving as a promotional item a "punch," which is a small bar of soft steel used to strike the intaglio die for their gravure typesetters. The punch is one-sixth of an inch on a side and contains the entire text of the Lord's Prayer, plus, of course, their company name and address. Each letter (mechanically carved) is 0.005 inch from top to bottom. A typical pixel on a 240 by 240 laser printer is 0.007 inch from top to bottom.
6. R. K. deBry, B. G. Platte, C. L. Berinato, and J. W. Marlin, "Architectures of Advanced Function Printing," *IBM Systems Journal* **27**, No. 2, 234–245 (1988).

7. *Presentation Text Object Content Architecture Reference*, SC31-6803, IBM Corporation; available through IBM branch offices.
8. *Image Object Content Architecture Reference*, SC31-6805, IBM Corporation; available through IBM branch offices.
9. *Graphics Object Content Architecture Reference*, SC31-6804, IBM Corporation; available through IBM branch offices.
10. *Bar Code Object Content Architecture Reference*, S544-3766, IBM Corporation; available through IBM branch offices.
11. Presentation text object is an exception. The parameters local to the text areas are contained in a presentation text descriptor in the active environment group.
12. *Font Object Content Architecture*, S544-3285, IBM Corporation; available through IBM branch offices.
13. Fonts were an exception because of royalty licensing and their enormous volume: 75 megabytes for one font in four faces and 14 sizes. Outline fonts have eliminated both hurdles: by reducing the library size by a factor of five and by breaking the royalty paradigm in which each rasterization carries royalty.
14. The AFP API is an interface permitting programmers of high-level languages to create MO:DCA documents from a COBOL or PL/I program.
15. "Edit" is a recent synonym for "redact," first used in this sense in 1791. "Redact" was first found being used in this sense in the fifteenth century. It is nearly time for a new definition for "redact."
16. 1982: IBM 3340-B1, $13,510, 34.9-megabyte capacity. 1991: IBM 3380-CJ2, $70,000, 1260-megabyte capacity.
17. "Mass Storage: Technology Overview," *Datapro Reports on Microcomputers*, McGraw-Hill Book Co., Inc., CM19-005-088 (August 1991).
18. "Document Imaging Systems and Software: Overview," *Datapro Reports on Document Imaging System*, McGraw-Hill Book Co., Inc. (February 1992).
19. N. J. Belkin and W. B. Croft, "Information Filtering and Information Retrieval: Two Sides of the Same Coin?" *Communications of the ACM* **35**, No. 12, 29–38 (December 1992).
20. C. A. Plesums and R. W. Bartels, "Large-Scale Image Systems: USAA Case Study," *IBM Systems Journal* **29**, No. 3, 343–355 (1990).

Accepted for publication June 7, 1993.

Roger K. deBry *Pennant, the IBM Printing Systems Company, 6300 Diagonal Highway, Boulder, Colorado 80301 (electronic mail: debry@vnet.ibm.com).* Dr. deBry is a Senior Technical Staff Member and manager of Advanced Technology in the Boulder Programming Center of the Pennant Systems company. He received his Ph.D. in computer science in 1972 from the University of Utah. Prior to his assignment in Boulder, Dr. deBry worked in the IBM laboratory in Kingston, New York, as manager of the IBM 3270 architecture group, and in the IBM headquarters facility in Purchase, New York, on the Engineering, Programming, and Technology Staff and the Corporate Programming Staff. He is a member of ACM, the U.S. representative to the IFIP Technical Committee on Programming, and a member of the IBM Academy of Technology.

Michael W. Munger *Pennant, the IBM Printing Systems Company, 6300 Diagonal Highway, Boulder, Colorado 80301.* Mr. Munger is manager of Performance and Architecture in the Boulder Programming Center of the Pennant Systems company. He joined IBM in 1968 as a programmer in the IBM Boulder laboratory. Beginning in 1978, he developed the Revisable-Form-Text Document Content Architecture. He moved to Austin, Texas, in 1981, where he managed the creation and development of MO:DCA. Since then, his responsibilities have included MO:DCA-P, IPDS, PTOCA, BCOCA, and FOCA.

Reprint Order No. G321-5528.