MULTIMEDIA PUBLISHING FROM A DATABASE

Michael O. Petrea and Kent M. Taylor

Michael O. Petrea *is a senior member of the information systems staff in systems design and integration, and* ***Kent M. Taylor*** *is a manager of strategic planning and systems support. Both are with the AT&T Network Systems' Document Development Organization (DDO) in Winston-Salem, North Carolina. Mr. Petrea is responsible for advance planning and systems engineering for DDO's electronic publishing systems, while Mr. Taylor's responsibilities include long-range business and technical planning, and systems design, development, and support for DDO. Mr. Petrea holds a Bachelor of Engineering Technology degree in computer electronics from the University of North Carolina at Charlotte, and an M.S. in computer science from the University of Illinois at Champaign-Urbana. Mr. Taylor holds a B.S.E.E. from North-*
(continued on page 71)

Technological changes have resulted in market pressures to deliver information products in multimedia formats. Such varied formats require more complex technologies than those needed when the producer is limited to traditional hard-copy media. Paper, still the dominant publishing medium, is being supplemented—and in some cases, supplanted—by Compact Disk–Read Only Memory (CD-ROM) on-line knowledgebases, and other forms of electronic information. Publishing databases must evolve to support all media from a single source description. This paper describes the evolution of single-source electronic publishing using the experience of AT&T's Document Development Organization (DDO) as a paradigm. For an overview of DDO, refer to Panel 2 of this paper.

Introduction

Traditional publishing has concentrated primarily on the various ways to deliver information in printed form. The computer-aided aspects of publishing have focused on more effective ways to collect and process information, up to and including pre-press production, i.e., the production steps (composition, pagination, page makeup) preceding plate making. The results have provided better, cheaper, and faster ways to get information into print.

But the marketplace continually demands more for less. In today's publishing world, this phenomenon is beginning to manifest itself through increasing demands for more efficient, cost-effective ways to deliver information. Delivery mechanisms, such as on-line knowledgebases and Compact Disk—Read Only Memory (CD-ROM), are examples of products gaining wide acceptance in the marketplace. This is not to say the print media will be entirely replaced by high-tech delivery systems. New delivery methods will differentiate the information product from its print-media predecessors.

From Cold Type to Procedural Markup

DDO's electronic publishing environment is the result of migrations to higher levels of computerization in its publishing processes. Since the early 1960s, DDO has embraced the available technology to the extent necessary to better meet the needs of its clients and AT&T's customers, and to remain both competitive and productive.

Earlier Attempts at Automation. Before the 1960s, composition and typesetting services had been purchased from outside vendors. By the mid-1960s, however, the volume of pages produced had grown so large that serious consideration was given to developing internal processes to support DDO's requirements. Thus, DDO began developing its own system, the Automated System for Composing, Revising, Illustrating, and Phototypesetting (AutoSCRIPT). AutoSCRIPT entered production in 1966, running on an RCA SPECTRA™ computer. AutoSCRIPT, a macro-driven text composition program, generated a *mile* of punched paper-tape output per day. (A *macro* is a program or script that executes several separate commands as a single operation.) The tape was read by a PHOTON-560™ second generation phototypesetter which imaged an entire text page in about 12 minutes. This was a true second generation machine, employing electromechanical technology as opposed to its purely mechanical first-generation predecessors.

In 1969, we introduced the RCA VIDEOCOMP-830™, a third-generation cathode ray tube (CRT) technology typesetter, into our production process. This typesetter was driven with magnetic tape and was considerably faster, producing about one text page every 12 seconds. Graphics were still produced manually, either photographically or "on the boards."

By 1975 we had upgraded to a new system, the Autologic APS-4™ phototypesetter. The APS-4, also a third-generation phototypesetter, was required to support phototypesetting of line art. In 1976 we introduced the Machine Aided Graphics for Illustrating and Composing (MAGIC) system, a vector-based 2½-dimensional illustrating system, developed by AT&T's Engineering Research Center in Princeton, New Jersey. Line-art creation was then moved from the drafting board to the MAGIC console, where the art was drawn by the console operator, then stored and managed in a database. (Currently, the MAGIC-2 system database stores over 300,000 drawings.) When needed for publication, the

Panel 1. Terms and Acronyms in This Paper

AutoSCRIPT	Automated System for Composing, Revising, Illustrating, and Phototypesetting
CALS	Computer-Aided Acquisition and Logistics Support system
CD-ROM	Compact Disk-Read Only Memory
DDO	Document Development Organization
document	A managed set of generic data downloaded to a production system for formatting and creation of a reproducible "master"
DOD	Department of Defense
EPS	Electronic Publishing System
EDDS	AT&T Electronic Document Distribution System
hypermedia	Refers to the integrated use of information from a variety of sources—text, graphics, data, and voice—as part of a hypertext system in which the separate forms are linked so the user can move between them
IDS	AT&T Information Delivery System
macro	A program or script that executes several separate commands as a single operation
MAGIC	Machine Aided Graphics for Illustrating and Composing
RBOC	Regional Bell Operating Company
SGML	Standard Generalized Markup Language
WYSIWYG	What you see is what you get

Panel 2. The AT&T Document Development Organization

The Document Development Organization (DDO), AT&T's primary in-plant publishing operation, is concerned with both technological and market pressures that affect information products. Concentrating primarily on AT&T product support documentation, DDO provides a complete range of publishing services, including documentation project planning and management; writing; editing; foreign language translation; composing and typesetting; image scanning and processing; and full service photographic and printing facilities. DDO also has one of the largest creative design and graphic arts groups on the East coast.

DDO's primary production facilities are located in Winston-Salem, North Carolina, with a local staff of about 800 employees. It also provides writing and other documentation services to on-site clients at over 20 AT&T locations worldwide. Virtually an AT&T in-house publishing company, DDO produces over 350,000 *original* pages per year, and in 1988, also produced over 150,000 electronic "pages." Growth in hard-copy production has been incremental while electronic output has doubled each year for the last four years.

illustration was typeset on the APS-4 phototypesetter. Both text and line drawings could now be typeset electronically, but not merged on the same page. The last step of actually merging the drawing and text remained a manual operation. Nevertheless, the level of automation achieved was preferable to the manual methods used previously.

By the early 1980s, DDO had again upgraded its typesetting capability with an Autologic APS-5 phototypesetter. AutoSCRIPT had evolved into AutoSCRIPT-2, running on a Digital Equipment Corporation DEC System-10™ mainframe. During this period, the use of centralized, shop-controlled batch composition peaked. Manuscripts were authored, edited, and sent to the AutoSCRIPT shop for composing.

The UNIX System Production Model. The introduction of the UNIX® system, with its Documenter's Workbench® environment and `nroff` and `troff` text formatters, marked the start of true computer-aided publishing at DDO, and continues to support about 70 percent of DDO's composed-page output. Our existing `troff` production capability, based on a group of centralized minicomputers, is actually used in a decentralized manner. The production model closely fits the classic definition of "personal" or "desktop" publishing, an environment in which users exercise control over all aspects of content creation and production.

This production model evolved during the "entrepreneurial" years following divestiture to meet perceived needs for "individuality" or "uniqueness" across the thousands of document projects supported by DDO each year. During this period, uniformity of documentation style all but disappeared, partly because the flexibility of `troff` itself provided users with an unlimited ability to customize their products. Furthermore, the UNIX system and its text processors became almost a standard within AT&T. As a result, user demands for more formatting capability, storage, and computer resources increased, as writers, editors, and input operators and compositors become familiar with and increasingly used the Documenter's Workbench facilities.

With sufficient hardware support, DDO achieved a successful, though limited, form of desktop publishing. Using `troff` alone, a user can input and appropriately code a document, format it, and receive composed output, either on paper or on a bit-mapped display, on a terminal such as the Teletype 5620 or AT&T 630 MTG. If the results are unsatisfactory, edit/code/compose

iterations can be repeated until the product is acceptable. In this environment, each document creator can input and tag text specifically for a single production process, e.g., `troff`, with a particular product output in mind, e.g., hard copy.

In the hard-copy environment provided by the UNIX system, instructions to the `troff` processor are based on the notion of *procedural markup*. That is, if a word requires an *italic* highlight, font change instructions are added to the passage for interpretation by the `troff` processor. These keystrokes (e.g., `\f2`, `.I`, or `\fI`) indicate a specific *procedure* that is open to no other interpretation by `troff`. Though this operational model adequately serves a single-product world, it is not an optimal solution when the user intends to produce *multimedia* deliverables such as electronic knowledgebases or voice products in addition to hard-copy output. That is, in order to produce deliverables for more than one type of output format or form, the content elements of a document must be identified with *generic names* (i.e., *tags*) rather than with *procedure names* on how to process them.

Adding Generic Markup: Evolution and Revolution

To provide additional composition capacity and functionality, DDO introduced several production-grade WYSIWYG (what you see is what you get) systems in 1986. These systems were supplied by Xyvision, Inc., and were installed along with a fourth-generation laser typesetter, the APS-6. The intent was to enhance our ability to produce high-quality composition, exercise greater control over page aesthetics, and allow the inclusion of scanned images for both line art and half-tones. The interactivity of the Xyvision systems was viewed as superior to the batch capability of `troff`. It quickly became clear that a new document markup language would have to be employed, because, like `troff`, the Xyvision systems would speak only their own language. Even more critical to our thinking was the advent of *multimedia* products for which DDO would be responsible. These products would require solutions to the problem of concurrent production in hard-copy and other (i.e., electronic) media.

As noted, procedural markup is useful only in the particular production environment to which it applies. Thus, if the product is intended for more than one distribution medium—e.g., electronic as well as paper—the developers face a choice. They can produce two entirely different versions using two separate forms of procedural markup to identify the content elements; or they can prepare *one* source document using *generic markup*, i.e., a form of code that can be interpreted differently by specific production systems to output the product in the desired medium. The production system "reading" the generic markup does so by interpreting the markup keystrokes via *style sheets* that define the formatting rules applied to the "marked" elements.

Generic markup, then, is sensitive only to structure, rather than form, and is independent of process. If DDO could institute generic markup to replace the more familiar procedural markup, new systems such as Xyvision could be introduced with fewer repercussions for the people generating input for them. We did this by developing the Electronic Publishing System (EPS) markup set. Refining this set resulted in a *generic macro package* that uses document style sheets driven by the EPS generic codes. The style sheets are developed, maintained, and modified by a centralized development group within DDO, rather than by individual writers and editors.

Our experience at DDO has reinforced our belief that document content should be stored and managed in a process-independent way. Indeed, the ability to produce multimedia documents and easily drive multiple production systems depends on process independence.

From Procedural to Generic. Markup methods at DDO have evolved with the UNIX system's `nroff` and `troff` formatters. The markup in use can range from "primitive" calls to the formatters, through user-defined macros, to defined and maintained macro sets, or can combine various combinations of methods. The result is

a collection of highly refined, optimized ways to format hard-copy using `nroff/troff` facilities. Unfortunately, this collection has built-in limitations, most notably the flexibility of `troff`. This flexibility has made it possible for groups—even, in some cases, individual users—to customize document markup and production to such an extent that it becomes difficult to share document inputs between or among groups. While everyone uses macros to one extent or another, groups tend to do so according to local conventions. This is similar to how a common language can be spoken in a variety of dialects that may make it difficult for one group to understand another. In such an environment, it is often difficult to use document content as source for input to other production systems.

DDO's production model for the Xyvision systems envisioned a division of labor in the document production process. That is, the content creation and production components of the overall document production process would be kept separate. A document would be input with generic markup, and then downloaded into a Xyvision system where production of the job—electronic page makeup—would be completed. The writer would thereby be freed from time-consuming formatting tasks. We have observed that separation of the writing and formatting tasks leads to greater productivity in both areas.

However, in the DDO environment, writers accustomed to the `troff` production model tended to react negatively to the introduction of the Xyvision systems. They felt the Xyvision approach—because they could not do their own formatting—would cause them to lose control of the total job. Further, it would relegate them to a "writer-only" role while at the same time they retained total responsibility for the customer's finished product. Most of all, the Xyvision approach would require them to learn an entirely new set of document markup conventions.

Our experience in migrating from one production model to another has highlighted the importance of seamless integration among systems. Users can and will change their operational methods if they perceive the change is in their best interests. This aspect is both an operational and a system design consideration. For any new technology to be accepted, the people who will use it must first "buy into" it as a valuable improvement.

Planning for Multimedia Products

Multimedia products result from the interwoven combination of market demand and technological growth. That is, what *can* be done is what the market *wants* to be done; and what the market perceives as a want or need becomes a priority item for technical development. In the following section, we discuss market support as a key function of electronic publishing organizations, and introduce the types of media that DDO is developing for product output.

Supporting the Market. Clearly, the market is becoming increasingly aware—and demanding—of alternate ways to deliver documentation, i.e., products delivered on media other than paper. DDO is already feeling the effects of this market demand. Besides hard-copy products, our market for on-line knowledgebases and conditioned data stream (i.e., screen-oriented text-displayed) documents is rapidly expanding. This growing need to deliver multimedia product documentation demands that we develop a production environment which is *not* biased toward a particular form of output product. The initial multimedia products to be supported are as follows:

- *Hard-copy.* Conventional hard-copy will continue to be a primary information delivery medium. Indeed, there are indications that hard-copy market demand may increase as users become aware of the value of information via other means.
- *On-line knowledgebases.* The AT&T Information Delivery System (IDS) is our system of choice for interactive on-line information delivery. This particular delivery method, employing hypermedia techniques, has gained wide acceptance within AT&T, and deployments are continuing an upward trend. (*Hypermedia* refers to the non-linear use of information from a vari-

ety of sources—text, graphics, data, and voice—as part of an integrated *hypertext* system in which information is associatively, rather than linearly (i.e., "start-to-finish"), linked.

- *Compact Disk–Read-Only Memory.* A commercially-available CD-ROM delivery system, compatible with AT&T IDS, is currently in use where portability is required. A PC-based version of IDS will be available this year.
- *Conditioned data stream (text only).* DDO currently delivers document files conditioned for display on customer-owned systems. At present, this is a text-only product, though plans exist to upgrade to full text and graphics capability. DDO has been approached by several Regional Bell Operating Companies (RBOCs) about this service.
- *Electronic page images (bit-mapped or coded).* Fully composed pages (text and graphics) will be delivered electronically to the AT&T Electronic Document Distribution System (EDDS) and/or other comparable demand-printing systems. The engineering and development effort to apply an EDDS architecture at AT&T's Customer Information Center is well underway. This activity will support the notion of "just-in-time-printing," i.e., producing output only on demand, thereby reducing on-shelf stock requirements.

Some Basic Assumptions. Our efforts to establish multimedia publishing capability is grounded in several assumptions:

- Market demand will require us to deliver product and service information on different media. Indeed, this has already happened.
- Writers will become increasingly responsible for content creation and less responsible for production. We believe this change is both necessary and desirable.
- A data capture system will be used to support content creation and other writing/editing activities. Data capture methods, such as image scanning and optical character recognition, will support the rush to "digitize" existing documents, without re-keying.
- A single production system will exist for each medium (e.g., hard-copy various electronic products).
- Operators trained on specific production systems will produce reproduction masters.

Building the Publishing Database

To meet these requirements for multimedia documents and products, DDO has developed and is implementing production from a *publishing database* of content-only information stored in generic form. This database allows document content to be stored and managed to avoid any emphasis on a particular type of medium or output. This "format-free" database storage will prove to be a useful and necessary design feature as new kinds of product outputs undoubtedly will need to be added in the future.

The publishing database will be our source of generically tagged information that can be downloaded into the various production systems designed to support specific products based on a single source document. The content of the source will have no form-related markup referring to any specific type of product. All materials—including graphic and tabular material—will be stored with generic markup.

Packaging and structuring to create alternate media information products will become a production process in its own right. The attributes of "form, look, and feel" will be added to each product at production time by operators trained in the aesthetics of the particular product (e.g., layout and typography, or interactivity and linkages).

The Conceptual Database System

Before we discuss the specifics of database publishing, a definition is in order. A *database* is simply a managed collection of stored, structured information that provides ready access to the information being managed. It can be viewed as a *management system that operates on a structured set of attributes that describe a collection of stored data elements.* The database manager allows the

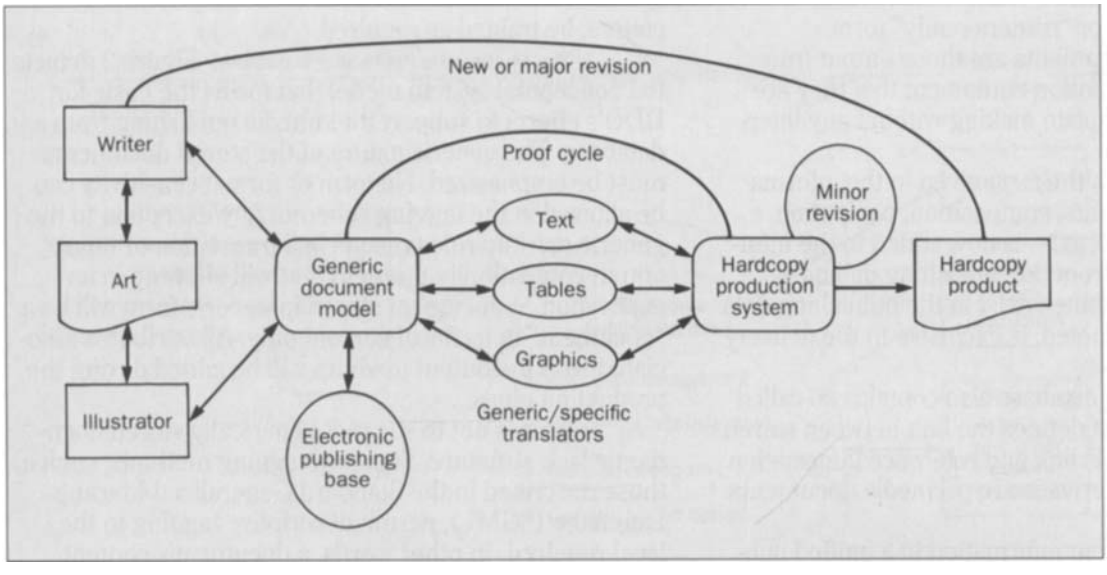

Figure 1. Hard copy production.

stored data elements to be accessed arbitrarily, i.e., in ways unrelated to the storage structure. Depending on the specific user requirements, it can also provide different *views* of the underlying structure of the data.

The Database Publishing Concept. The primary difference between database publishing and more traditional applications of computer-aided publishing is in the way information is stored. In traditional, print-directed systems, the stored information tends to be highly format-oriented. However, a publishing database used to drive multimedia output products should endeavor to be generic, free-form, and flexible. That is, the stored information should be independent of both format and process.

In reality, we are really speaking of *two* databases: a process database and a product database.[1] They are differentiated as follows:

- The *process database* is the base of working information that serves as the source data for various output products. That is, it contains the words and generic codes (i.e., the *content model*) from which multimedia outputs can be produced. It is strictly *source*, and cannot be turned into a product without some form of interpretative style sheet or formatting process. A publishing database, then, is a process database that contains information useful for publishing derivative products. It is useful only for managing *information* from which derivative products are produced, i.e., it is not itself directly capable of producing a "product."
- The *product database* is itself a form of output product, e.g., an on-line knowledgebase accessible by a user. It contains the information required to "read" the generic input in the source file and turn it into a product.

A source document is defined and downloaded into a production system. The particular form of the product is added and output is generated. For instance, if one wishes to produce a hard-copy document (Figure 1), the following process would apply:

1. The content model of the document is used to extract a subset of the information stored in the publishing database.
2. This information is downloaded into a publishing system (i.e., a *product database*) designed to provide a

document in typeset or "camera-ready" form. ("Camera-ready" documents are those output from sufficiently high-resolution equipment that they are considered ready for plate making without any intervening steps.)

3. Additional value—i.e., the *form* given to the information being typeset (fonts, composition, pagination, color, art placement, etc.)—is now added to the information downloaded from the publishing database.

None of the formal attributes exist in the publishing database, because form, as noted, is exclusive to the delivery media.

The publishing database also contains so-called "hyper" information that defines the link between stored information entities. This link and reference information provides the basis for derivative hypermedia documents or knowledgebases.

Storing all content information in a unified publishing database reduces the proliferation of duplicated source information. The information can be downloaded from the publishing database into as many production systems as needed. Once downloaded, a *production image* of the stored content information exists in the publishing database. Editorial and text changes to the product output should be avoided at this point, because content changes should be made only to the content model stored in the publishing database. In the software world, this is analogous to changing the source code rather than using binary patches.

In practice, a word processing environment is used as a "front end" to the publishing database. Production system operators then add product-specific "form" to the document content, thereby creating the finished product. Thus, subject matter experts create content, production experts create products. Output processes (that is, "products") can be added or deleted with little or no effect on the word processing front end, resulting in reduced training requirements for most system users. Operators of the various production systems will, of course, be trained as required.

The Multimedia Publishing Database. Figure 2 depicts the conceptual system model that forms the basis for DDO's efforts to support multimedia publishing from a database. The generic nature of the stored documents must be emphasized. No form or format sensitivity can be allowed in the tagging scheme. Any exception to the generic markup rules results in a translation or input/output compatibility problem that will show up in an application. A document stored in generic form will be a "document" in terms of content only. All attributes associated with the output medium will be added during the production phase.

This is not to suggest generically stored documents lack structure. Standard tagging methods, such as those described in the Standard Generalized Markup Language (SGML), permit descriptive tagging to the level required. In other words, a document's content components can be identified for location, retrieval, validation, or any other database manipulation. Such a storage scheme easily allows the inclusion of linking mechanisms to support hypertext.

With the markup language, text representation issues are reasonably solved. However, documents contain other content entities as well. Items—such as tables, vector or image graphics, and equations—often are represented in system-dependent ways. This makes it difficult to describe these content entities generically to permit portability. While there are practical—if nonstandard—solutions available through the UNIX system (e.g., `pic`, `tbl`, and `eqn`), for the moment, portability issues remain. They will disappear as standard representations for nontext entities are developed.

Figure 2 emphasizes our restructuring of responsibility, i.e., the burden of production will shift from the writer to the designer or formatting specialist. This change will make it easier to add new production processes without having to retrain the writing staff. While operators for a particular production system will have to

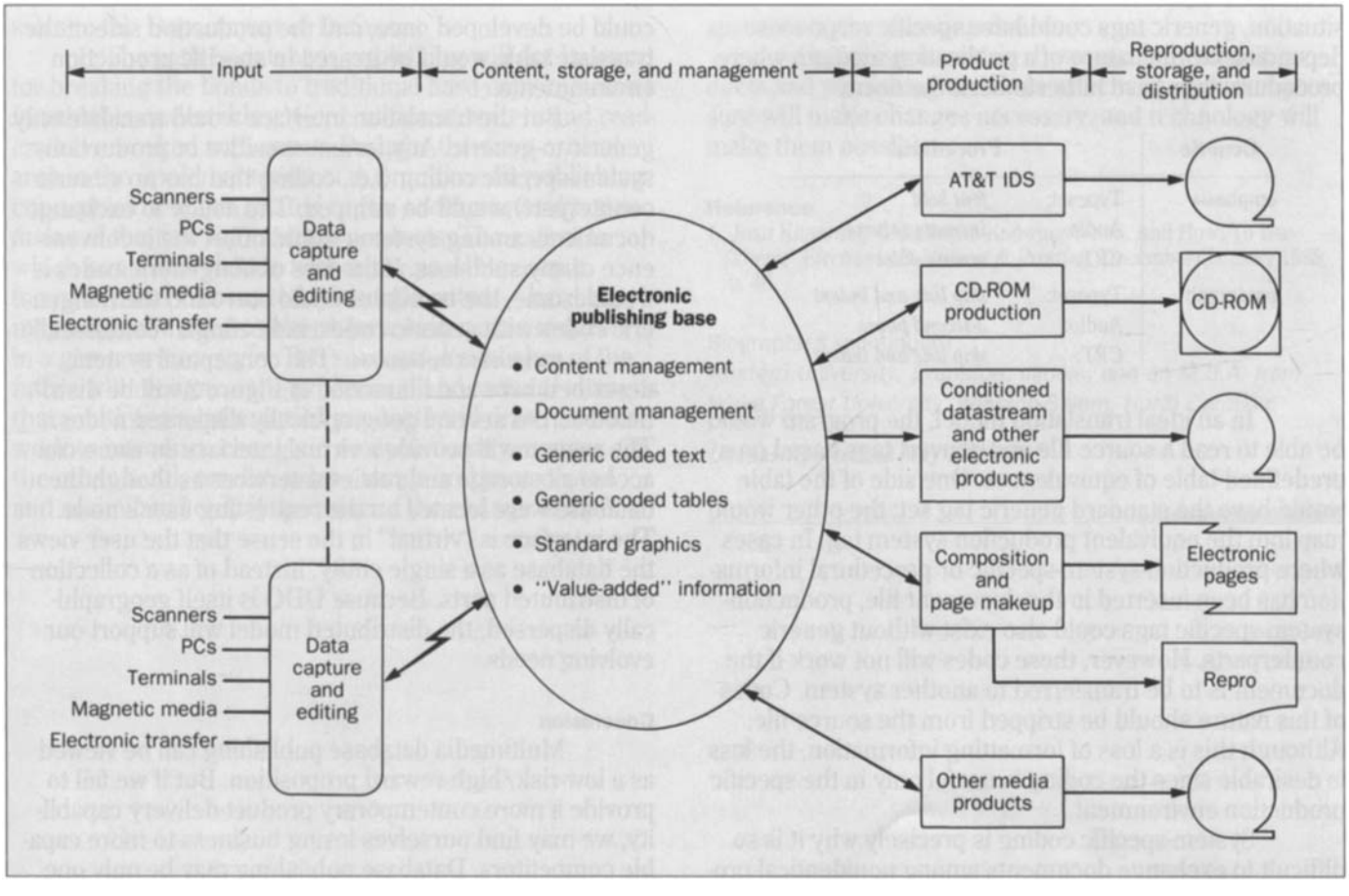

Figure 2. Multimedia publishing concept built on an electronic publishing base.

be trained, one production system operator will be able to support the work of many writers, reducing training requirements while increasing efficiency.

Two-Way Translations. Figure 2 shows that translators are critical connections between the publishing database and the production systems. Information must be portable, requiring translators to work well to achieve seamless integration. If generic markup concepts are followed, each system need not use exactly the same markup. Though syntactically different tags are used, tags that identify unique stylistic elements should translate on a one-to-one basis. Once a future standard set of generic tags is adopted, a relationship can be defined between the standard tag and the equivalent production system tag. For example, in a multimedia publishing

situation, generic tags could have specific responses depending on the nature of a publication medium where procedural tags have hitherto been the norm.

Generic	Procedural	
emphasis	Typeset:	*font bold*
	Audio:	*increase volume*
	CRT:	*reverse video*
paragraph	Typeset:	*skip line and indent*
	Audio:	*3-second pause*
	CRT:	*skip line and indent*

In an ideal translation model, the program would be able to read a source file and convert tags based on a predefined table of equivalences. One side of the table would have the standard generic tag set; the other would map into the equivalent production system tag. In cases where production-system-specific or procedural information has been inserted in the document file, production-system-specific tags could also exist without generic counterparts. However, these codes will not work if the document is to be transferred to another system. Codes of this nature should be stripped from the source file. Although this is a loss of formatting information, the loss is desirable since the coding is useful only in the specific production environment.

System-specific coding is precisely why it is so difficult to exchange documents among nonidentical production processes. Such coding should be avoided, at least in files that are to be interchanged with other systems. However, within a production system, system-specific codes will surely have to be employed to produce formatted products.

The table translator in our ideal model allows translation in *either* direction. The advantage of translating to a standard form is that only one bidirectional translator per participating system is needed. Since the table look-up program is not unique to a particular system, it could be developed once, and the production side of the translate table would be geared to specific production environments.

But the translation interface would translate only generic-to-generic. Any format-sensitive or production-system-specific coding (i.e., coding that has no generic counterpart) would be stripped. The ability to exchange documents among systems would offset the inconvenience of any such loss. If the loss of nongeneric codes is troublesome, the translator could surround the nongeneric codes with generic codes, indicating a "comment."

Distributed Operations. The conceptual system described here and illustrated in Figure 2 will be distributed across several geographically dispersed nodes. The system will provide a virtual interface so users can access all storage and retrieval services as though the database were located on the requesting user's node. The interface is "virtual" in the sense that the user views the database as a single entity, instead of as a collection of distributed parts. Because DDO is itself geographically dispersed, the distributed model will support our evolving needs.

Conclusion

Multimedia database publishing can be viewed as a low-risk/high-reward proposition. But if we fail to provide a more contemporary product-delivery capability, we may find ourselves losing business to more capable competitors. Database publishing may be only one resolution to the issue of providing consistent information content in multimedia form. Certainly, it reflects a growing trend in the publishing industry.

Adding support to our design rationale is the Department of Defense (DOD) Computer-Aided Acquisition and Logistics support (CALS) initiative. With CALS, DOD requires its suppliers to deliver product documentation electronically using SGML. With the appropriate translators in place, DDO's system will be able to produce a CALS-compliant output that will support AT&T

sales to the Department of Defense.

The path to database publishing will not be easy, for breaking the bonds to traditional hard copy remains a formidable undertaking. Generations of writers and readers have worked with paper. Perhaps the most critical area in the implementation of the multimedia publishing concept is therefore related to the *packaging and presentation* of the new information products. The extent to which content is written with many possible output forms in mind (versus today's single output—hard copy) may suggest a need to alter how a writer uses *structure* in a given writing style. The successful publisher of the future will recognize the procedural and *cultural* changes that multimedia information products will drive, and will work to introduce change in an evolutionary manner. On the other hand, *revolutionary* change will be accepted and assimilated only if the value of the change is made apparent to the user population.

One thing is certain, however: information products and publishing methods will change. Market pressure will make changes necessary, and technology will make them possible.

Reference

1. Joan Knoerdel, "Databases: Knowing When, And How, To Use Them," *Electronic Publishing & Printing*, December/January 1988, p. 48.

Biographies (continued)

western University, Evanston, Illinois, and an M.B.A. from Wake Forest University, Winston-Salem, North Carolina.

(Manuscript received May 1, 1989)