THE GILBANE REPORT™

on Open Information & Document Systems

Vol. 1, No. 3
July 1993

Publisher:
Publishing Technology Management, Inc.
Applelink: PTM

Editor:
Frank Gilbane
fgilbane@world.std.com
(617) 643-8855

Subscriptions:
Carolyn Fine
carolyn@world.std.com
(617) 643-8855

Design & Production:
Catherine Maccora
(617) 241-7816

Associate Editor:
Chip Canty
ccanty@world.std.com
(617) 265-6263

Contributing Editor:
Rebecca Hansen
MCI Mail: 4724078

Document Management & Databases

There are many different types of systems being sold to manage documents. They range from electronic mail-based office workflow systems, to high volume document imaging systems, to compound document configuration management systems.

Some of these systems are designed primarily for one type of document, such as a form or an engineering drawing, and others are meant to be more general purpose. Over time we will analyze each type of system to help you determine which kind of solution is the best fit for different document management needs.

In this issue we focus on how database technology is used in what we would call "high end compound document management systems". These systems are characterized by the capability of dealing with large volumes of dynamic information, and the ability to manage the wide variety of text and graphic digital formats that make up today's electronic documents. Such systems are typically used in strategic document applications where data integrity and security are critical.

There are different database models suppliers must choose from, and there are a number of ways to combine document and database technology to build document management systems. Our article lays out the approaches taken by some of the major vendors and discusses some of the reasons why, to help you put your document management needs and expectations in perspective.

Coming In September! — Electronic Distribution

Our next issue will be devoted to sorting out the issues in choosing an electronic distribution and viewing solution for corporate publishing applications.

Contents

Document Management & Databases — What is the Relationship? ◀ Page 2

Documation '94 Update/Calendar of Events ◀ Page 18

Topics To Be Covered in Future Issues ◀ Page 19

DOCUMENT MANAGEMENT AND DATABASES — *WHAT'S THE RELATIONSHIP?*

Executive Summary

Strategic Overview

- Electronic documents contain valuable information. This information should be subject to the same level of management and protection traditionally provided by databases.
- Today's database and document system technologies provide the tools we need to begin capturing and managing document information in comprehensive and meaningful ways.
- These technologies are already being integrated into the first of a new generation of document systems that can provide sophisticated document-aware information management capability.

Document Management and Databases

- None of the common database architectures are optimized for documents, but third parties have built solutions on top of them that improve their document handling capabilities.
- Some store documents in relational tables, others store information about documents in tables that point to documents stored in files.
- "Textbases" use the document itself as a sort of database. They typically store documents in structured files augmented by indexing schemes designed specifically for document management and and retrieval.
- Organizations must determine whether they need primarily to store and retrieve documents or to manage the information *within* documents.
- File-based systems generally provide document-level management. Solutions that store documents in databases are often more efficient at managing each document's components.
- Component-level management can offer important benefits because it facilitates the reuse of information originating in documents — not just for other documents but for applications that may not involve documents at all.
- Component-level document management does, however, represent a substantial performance challenge for databases since they must be able to reconstitute documents as well as perform updates across a web of relationships that is far more complex than traditional transaction-oriented data.

Risks and Costs

- The largest risk in implementing a document management system is in underestimating the potential complexity of the information in documents and what you need to do with it.
- The exposure to risk can be minimized with a clear understanding of your immediate needs, and with a parallel plan for any related organizational or business process

changes. Performance, security, and interoperability with existing systems are the areas to watch out for.

- With the technology moving rapidly, organizations risk making choices that limit their flexibility by focusing too narrowly on meeting their current document management needs, and thus ending up with solutions that may not have the capacity to expand as their needs grow and change.

Recommendations

- Begin by analyzing your workflow and documents. The effectiveness of the solution will depend to a great extent on how well your organization understands its own requirements and the quality of the data and process models you develop to represent them.
- A document/information re-engineering exercise can help you choose between document management approaches. The three approaches described make very different assumptions about document processing needs.
- Pay careful attention to the way you will be storing and accessing information. If you want your information to be 'open' consider the benefits of SGML. If you want to freely access the information make sure that the flavor of SQL, or other document query language you will be using, is widely supported. Also, be sure that the query language is capable of meeting your information access requirements.
- Identify a document management solution that offers the simplest possible way of doing the work you do today, while leaving you options for the future. If you're committed to a particular database, look at solutions that build on the capabilities of that database. Consider a solution that makes use of a hybrid database or one of the textbase solutions if the additional implementation and support costs are compensated for by features and performance that are better attuned to your document management requirements.

Strategic Overview

Organizations are becoming increasingly aware that a great deal of their valuable information is in document form. And they're also increasingly uncomfortable with the fact that documents are scattered around the organization on all kinds of unprotected disks, locked into application-specific formats.

Ideally, we'd all like to subject our documents to the same rigorous security and control now exercised by relational databases on fielded data. We also need an industry-standard way of accessing it (see our article on Document Query Languages in *vol. 1 no. 2*). And we want fast access to our documents and the ability to join document information automatically to other types of data.

So why not just put our documents in a database? Unfortunately, none of today's databases were designed to handle them. Handling documents properly in a database requires some form of optimization — we must either invent a more appropriate database model or add new document-oriented tools atop existing database engines.

Fortunately the 1980's saw much progress toward the goal of managing document information more effectively:

- Publishing system vendors began moving beyond their preoccupation with composing pages to address issues involving the management of documents and the workflows of document creators and users.

- The Standard Generalized Markup Language (SGML) became widely adopted in some publishing-intensive industries as a means of encoding document content and structure so that it could be stored, retrieved, exchanged, and edited electronically, independent of the user's platform or application. For many companies, SGML will be the key to capturing information in reusable form (like reusable program code). But new or expanded database structures will have to emerge to accommodate this rich information.
- Imaging emerged as a means of storing and retrieving document facsimiles — and was later coupled with other technologies, such as OCR, that make it possible to search and retrieve information within documents.
- Full-text retrieval, another technology often coupled with imaging, became widely used and increasingly capable (with the incorporation of context-based and fuzzy logic capabilities) of focusing in on the information the user wants.
- Object-oriented processing models became commercially viable, with applications in user interface, software, and database design. The benefits of the object-oriented approach include the ability to operate on unlimited data types and to modify object-based designs easily over time.
- ANSI SQL became the standard language for querying relational databases.
- Organizations began modeling their workflows and information requirements as part of the effort to re-engineer business processes.
- Client-server architectures and more powerful platforms evolved to provide an appropriate environment for distributed access to document repositories.

As a result of these trends, database vendors today are under pressure to accommodate not only documents but the coming wave of multimedia and interactive information that will be published electronically. They are now working feverishly to expand the capacity of their products to handle a wide variety of data types.

There's also a great deal of collaboration going on between database developers and developers of publishing, text retrieval, and workflow software. As a result, isolated products will give way to increasingly integrated solutions and users will enjoy an expanding set of options for capturing document information and managing and retrieving it in flexible and meaningful ways.

Document Management and Databases

Today's Database Models

There are different kinds of database technology that can be coupled with document management applications. Is one model more appropriate for managing documents than another?

Relational Databases

Relational Database Management Systems (RDBMSs) such as Oracle and Sybase are the most commonly used form of data management. They were designed to handle information that can be broken down easily into tables and have been optimized for transaction-oriented applications in which small amounts of data is being retrieved and updated rapidly.

The initial applications of databases to documents have been for "database publishing," where the goal is to produce a document automatically from a database of information.

In most cases, small amounts of text are entered directly into database fields using a form-like interface. Information is then compiled, output to a file, and composed and paginated by electronic publishing software in batch processes that require little or no human intervention.

Broader applications for publishing from databases were stymied for years until RDBMS vendors extended their products to handle larger database objects. Binary large objects (BLOBs), variable-length records containing data in any format, can be used to store text or graphic objects of virtually unlimited length, including entire documents.

Still, there isn't universal agreement about whether it's a good idea to store documents as BLOBs. Depending on the length of the BLOB, performance can be an issue. In addition, depending on your application, the ability to store a document in a BLOB may not be all that useful. Since the RDBMS doesn't know what's inside the BLOB, it can't do anything with document content.

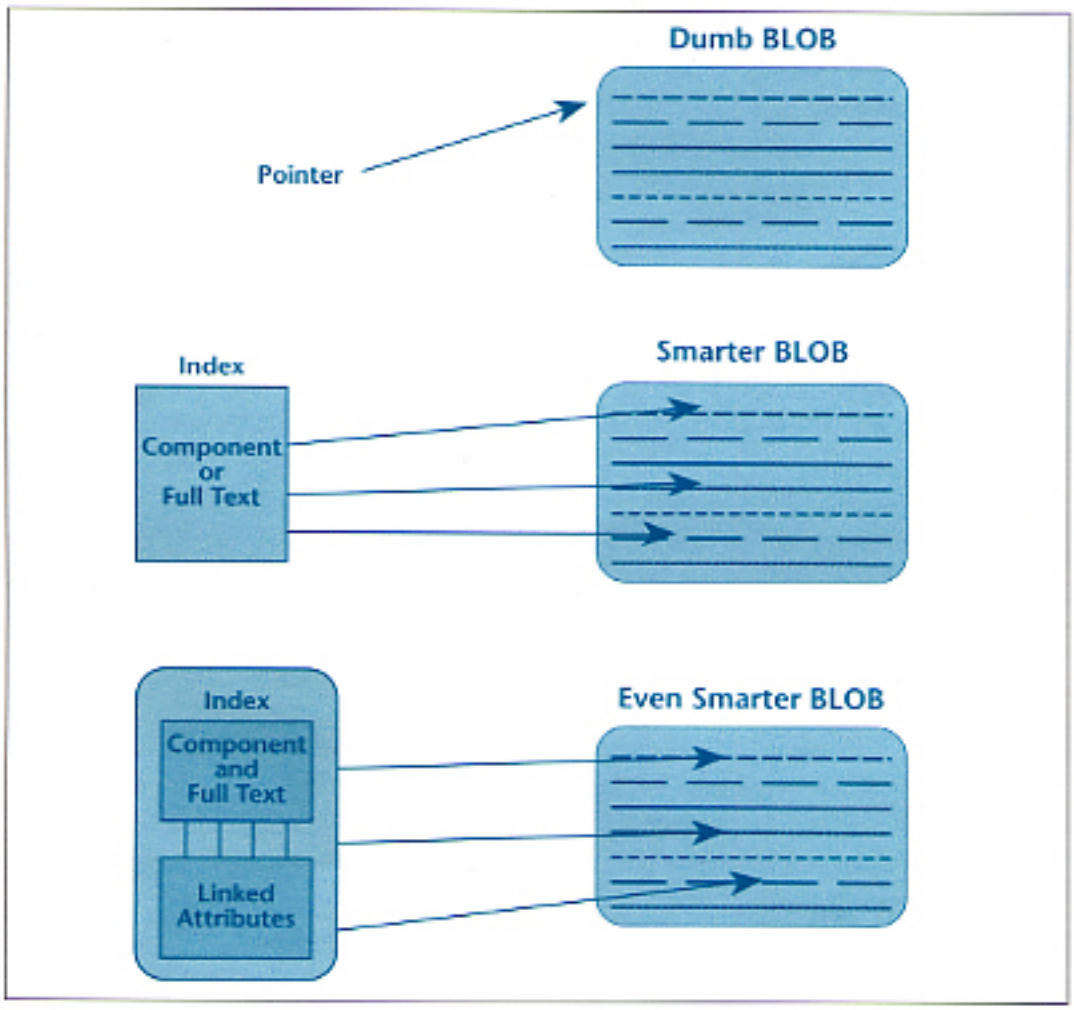

Figure 1

Not all BLOBs are created equal. Here are some of the possibilities

Some products have added the capability of indexing BLOBs for full-text search. To go beyond retrieval and really manage document information, however, you need a way of pulling structural and attribute information out of the BLOB and managing it in relational tables and/or of breaking down documents and storing them as smaller component-level as BLOBs. If you break down your document into smaller BLOBs (and there are some very good reasons why you might want to do that, as we'll discuss later), the database must

be able to reassemble the components back into the document. This, too, raises performance issues.

Object-oriented Databases

Object-oriented databases (such as Object Design's and Ontos's) are often proposed as better suited to managing documents than relational databases. Object-oriented concepts were designed to make it easier to develop and evolve complex systems and designs. As such, Object-Oriented Databases (OODBs) may have the potential to address the implementation of document data models, which are far more complex than the data models traditionally managed in the relational world. There are also potentially useful similarities between object-oriented technology and SGML. Both have a mechanism for associating information objects with attributes and both make use of the concept of hierarchy.

"Because relational and object-oriented database each have powerful capabilities, why not have the best of both worlds?"

Even so, OODBs have yet to be proven robust alternatives to relational databases and currently lack some key features database users have come to expect (*e.g.*, a powerful nonprocedural query language, automatic query optimization and processing, automatic concurrency control). It's also unclear how far the apparent suitability of OODBs for documents goes. Some developers who have experimented with using OODBs for documents say that OODBs are too general purpose to be really efficient. The fundamental OO principle of encapsulation (the hiding of object attributes and methods inside an object), for example, may not be practical when it comes to implementing a large complex document database since it involves a lot of overhead.

For now, let's just say the jury is still out. What is clear, though, is that OODBS, like RDBMSs, need to be optimized to handle documents.

Hybrid Databases

Because relational and object-oriented databases each have powerful capabilities, why not have the best of both worlds? Some vendors have long claimed to build an object-oriented schema atop a relational database model. But others are now taking this concept several steps further in new products being developed.

Information Dimensions Inc.'s BASISplus is an extended relational database that accommodates variable-length text. Their docXapi enhancement, now in beta test, provides an object-oriented document model and API on top of the relational storage manager.

IDI says this will enable users to specify a particular subset of object attributes (anything from "author" at the level of a document to "part number" at the level of a paragraph or assembly instruction) that they want to manage as explicit properties. Properties, which are indexed and searchable, have pointers back to the actual document content they were derived from, which is stored as a BLOB.

UniSQL is a hybrid database that melds relational and object-oriented database capabilities into a single layer. That is, UniSQL's relational tables incorporate the OO concepts of encapsulated attributes and methods as well as of inheritance. This provides some potentially powerful capabilities for managing documents (such as a ready means of capturing document structural relationships).

Still, Texcel, which is using UniSQL to develop document management solutions, says that while the database offers more of the tools they'd like to have for handling documents than do relational databases, there are still a lot of things they must do in implementation to support the full range of capabilities needed for documents.

Which Database Model Should You Choose?[1]

With no clearly superior database solution for documents, vendors of document management products have taken a variety of approaches. Most, but not all, make use of RDBMSs in some way — understandable since these are what most of their customers already have in place. There are at least three distinct approaches, however, with different implications for how you store and use documents:

- Storing information about documents in relational databases
- Storing entire documents in relational databases
- Using documents themselves as "databases" (or "textbases")

Solutions That Store Information About *Documents In Databases*

"The disadvantages of file-based systems are that they do not provide the powerful data security and integrity features of relational databases."

The initial approach of many companies has been to store "meta-data" or information about documents (*e.g.*, author, time stamp, version, security level) in the database with pointers to the actual documents stored in files.

This approach was dictated by the inability of relational database technology until recently to handle text. Many vendors continue to believe, however, that it's the most practical approach since users continue to think about and work with their documents primarily as files. Another advantage, especially in networked environments, is that the data about documents is generally more compact than the documents themselves. Consider, for example, one company that needs to provide access to the same documents to workgroups located thousands of miles apart. Sending documents back and forth over the organization's T1 satellite link would not be efficient; instead this firm maintains duplicate files at each location, while managing access to theses files through a central relational database.

The disadvantages of file-based systems are that they do not provide the powerful data security and integrity features of relational databases. You can, of course, augment the security features operating at the meta-data level in the relational database with additional security in the document file server, but it becomes fairly complicated since you're having to do it in two places. File-based systems also present more complications when it comes to wide-scale configuration management and functions that integrate document information with other types of information stored in databases.

Examples:

Interleaf RDM is a "document lifecycle" management product that combines library management, configuration management, and workflow. It uses an object-oriented document manager with a relational database (Oracle) that has pointers to document files.

By combining Interleaf's "active document" technology with RDM it is also possible to access and manipulate components within documents. This makes RDM most useful for those whose documents are in Interleaf format; however, RDM can also manage files created using other text and graphic applications.

Work Group Technologies CMS (Configuration Management System) is a document and workflow management solution that comes out of the engineering world. It enables companies to model, automate, and control product design approval and change

[1] This section describes several document management products in each category. It is not an all-inclusive list; the fact that we mention some products and not others should not be read as an endorsement. Our sole aim is to illustrate the range of document-management options available today and the different information-modeling approaches adopted by different vendors.

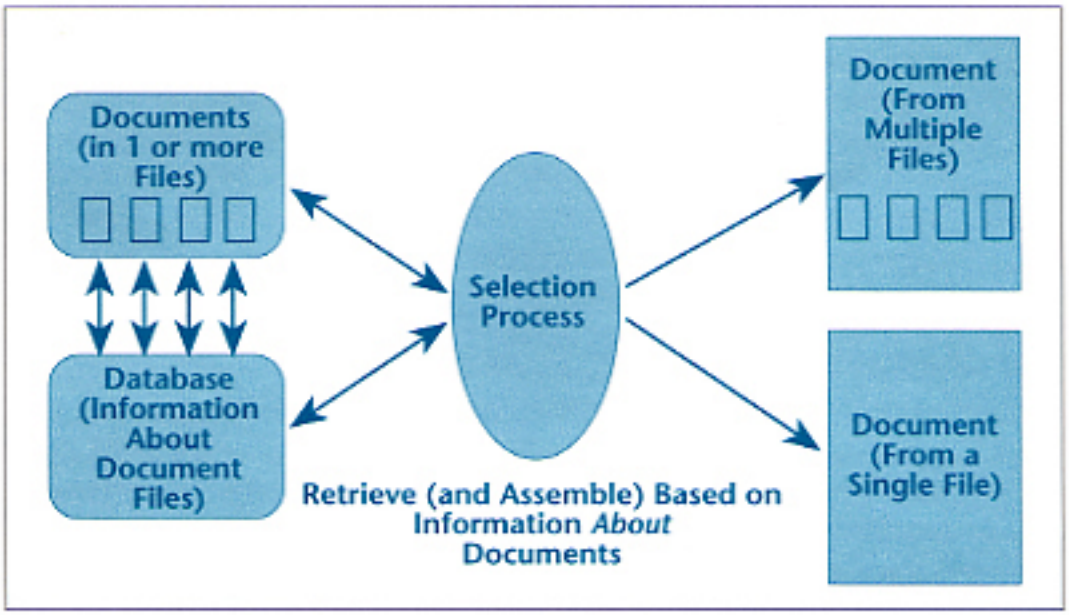

Figure 2 Using a Database to Manage Document Files & Information About Documents

processes as well as manage all related documents (*e.g.*, CAD drawings, technical manuals, parts lists, manufacturing process documentation).

CMS uses an Oracle or Sybase relational database to store information about documents and pointers to document files, which remain in their native formats. It, too, can manage any type of file.

Documentum is an integrated document management/workflow solution that provides object-oriented document services and a content manager that maps data attributes stored in an Oracle or Sybase database with document content stored in magnetic files or on optical discs. The content manager also automatically initiates indexing on specified information (the product integrates Verity's content-based retrieval engine) for full-text searches.

Solutions That Store Documents In Databases

Another approach is to store documents along with the meta-data inside the relational database. An advantage of this approach is that documents can be managed with the same level of security and integrity as other data.

The in-database approach can potentially provide a lot of flexibility for managing information not only at the file level, but at finer levels of granularity. To do so, however, the document management solution must provide some means of "shredding" documents into components and loading those into the database. Also needed is a querying method for retrieving components, and a mechanism for reconstructing shredded documents.

Examples:

Xyvision's Parlance Document Manager (PDM) is an information lifecycle solution that takes this approach and also provides tools for workflow management. SGML-tagged elements are stored as components inside of the relational database. (PDM currently requires documents to be in SGML to be managed at the component level.) PDM can also manage external files. PDM provides an easy graphical way of viewing database contents according to user-specified views. You can look at your data, for example, as a set of icons depicting document structure (documents, document sets, chapters, sections,

paragraphs). You can also look at document components according to author or work group, version level, project, etc. The cuts you can make through the database depend on how you design your data model, although PDM also lets you search for information that has not been specified as an organizing parameter.

IDI's BASISplus, as we noted, can store SGML structure and content inside of relational records while maintaining links to graphics, image, video, and audio files. BASISplus also incorporates automatic SGML parsing.

Odesta's Open ODBMS stores documents either as files or in BLOBS, in both cases at the document level (Odesta says that they have not had much demand yet from their customers for component-level storage or SGML). The product, which can make use of Sybase, Rdb, or (due out this summer) Oracle relational databases, provides a graphical interface that lets users manipulate document and workflow objects, with ODMS generating standard SQL queries in the background to retrieve information from the

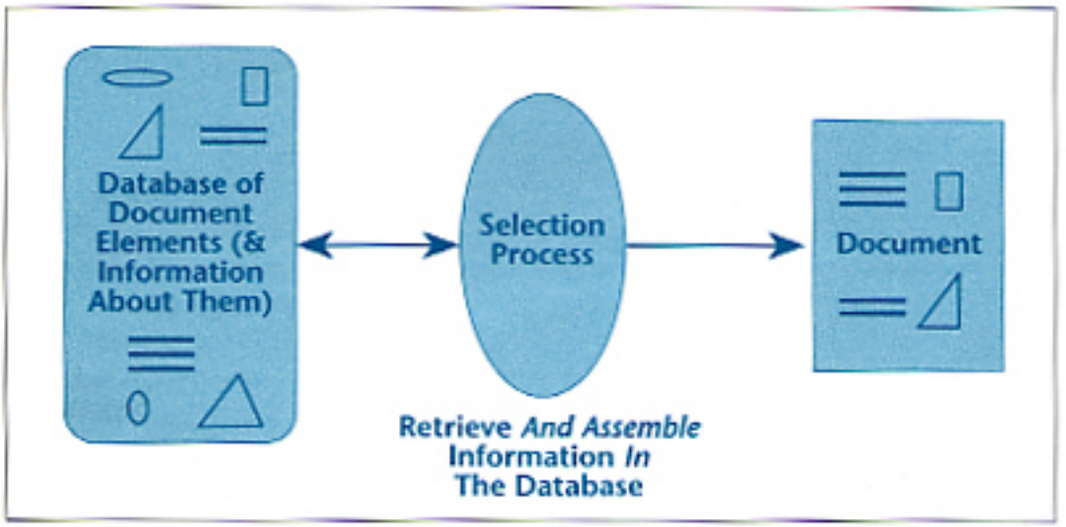

Figure 3 Using A Database To Store & Manage Document Information

database. (Standard SQL is adequate since Odesta is not trying to access the information within BLOBs.)

Texcel is currently developing a product, called Information Manager, which will make use of UniSQL's hybrid data storage capabilities while providing a data model optimized for SGML documents. Described as a "toolkit for building information management solutions," Information Manager is a server for managing SGML objects as well as the document meta-data needed for controlling workflow and other applications. The product will provide tools for creating database schema on-the-fly from DTDs, loading SGML document components into UniSQL, retrieving them using an "SGML query language," and managing them through both general and application-specific services.

Both Documentum and Interleaf's RDM, which currently store only meta-data in relational databases, plan to add the ability to store document components as BLOBs. Documentum says this capability will be available in their upcoming 1.1 release. Integration with an SGML parser is planned for the future. Interleaf, which already sells a parser with other SGML products, has not yet announced a delivery date for BLOBs.

Solutions That Use Documents As Databases

There are also "text bases" that have been built from "the ground up" with structured documents in mind. (This category raises many issues unrelated to the use of relational or

object-oriented databases, and will be covered more fully in a future issue where we will also look at non-structured text bases. The brief discussion here is included for completeness.)

The reasoning behind this approach is that documents (particularly SGML documents) have a natural structure and thus provide a readily available model for a database. Why break this structure down into rows and columns — why not just use it?

Generally these products leave structured (tagged) documents in files, and provide indices of one type or another to search, retrieve, and manage the information within the file. The advantages and disadvantages are similar to those that use relational tables to point to files — in particular, users need to make sure that adequate file security is provided and that performance is acceptable.

Examples:

Open Text supports both SGML and other structured files. It also provides rapid full-text and structural search capabilities.

OfficeSmith was designed just for managing SGML documents. It stores tagged text as data objects within a tree structure and uses indices to navigate the structure.

Berger-Levreault's SGML/DB maps SGML information into relational database structures to facilitate performance and security requirements for dynamic SGML element transactions.

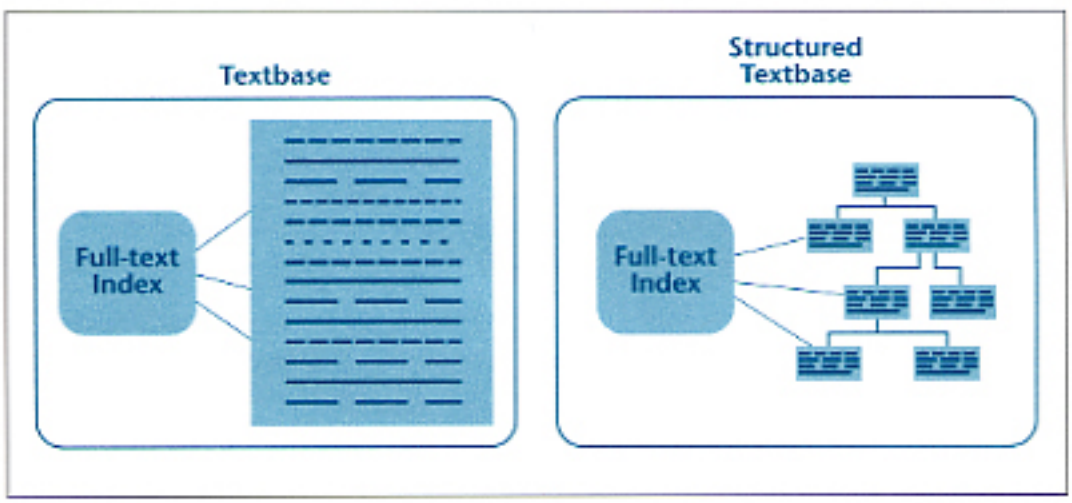

Figure 4 How Structured Textbases Differ From Full-text Databases

Questions To Consider

Before choosing a document management solution, it is important to consider all that is unique about your application, your documents, or your workflow. This section presents several questions you should consider before committing to a particular strategy or product.

What Do You Need To Manage?

A key distinction we keep returning to in describing these various approaches is the size of the chunk of information they are designed to manage.

Products that store only information *about* documents in databases are, generally, best-suited to managing whole documents, or chapters and sections of them — anything you would be likely to save as a file. Such products sometimes allow you to combine objects or manage several files as a single composite object, but you cannot manage objects within a file.

In contrast, products that store documents inside databases give you the opportunity, at least theoretically, to manage documents at the component level. As you'll note from the descriptions above, however, not all currently provide the tools you need to do that without a lot of systems integration work.

Products that use documents as databases, the so-called "textbases," let you access documents as files while also providing tools for managing the information components in the files. In general, however, they are more proven for search and retrieval operations than for other types of component-level management.

How Do You Work With Documents?

The choice between these approaches should be based on maximizing performance for the types of work you do most. For example, if you're working on catalogues or directories, you'll probably need to retrieve and update small chunks of information very rapidly. In this case, you'll clearly benefit from storing these chunks as BLOBs in a relational database. If, on the other hand, you work with files that must be retrieved in their entirety and retained for long editing sessions, a file-based system may be more efficient.

"The choice between these approaches should be based on maximizing performance for the types of work you do most."

Often the choice will not be so clear cut. Andersen Consulting's Pacific Northwest Document Management Group, is working with the Wisconsin State Legislature to design a document management system for lawmakers. Legislative documents are generally reviewed and edited in their entirety, and thus they are stored as files. At the same time, specific information in these documents needs to be extracted, stored in a database, and combined with other database information for automatic inclusion in the legislature's daily journal publication. In this particular case, Andersen is integrating an Interleaf 5 publishing system with a SmartLeaf "shredder" and Documentum for both file and database storage.

Many organizations have workflows that similarly encompass a mix of requirements. Take the preparation of a New Drug Application (NDA): The information that goes into an NDA comes from many different clinical and research departments, each with a unique purpose and way of interacting with the information. Some users may want to work with small chunks, such as drug description paragraphs; others with whole documents, such as clinical reports. At the end of the process there is typically a regulatory affairs department that receives the information in all kinds of forms and then has to piece it together.

This type of workflow — where people need to access information that originates outside of their own domain of work, but in a way specific to their own group or task — is one of the best arguments for component-level management. This is because it gives you comprehensive access control as well as the ability to manage information at the lowest common denominator. Of course, there may be some performance tradeoffs between the ideal unit of work that would be most efficient for a particular group, and the practical unit of reusability that is most efficient for the organization as a whole.

How Do You Deliver Information?

The questions that apply to authoring also apply to information delivery: Do you need to provide customers and other end users with access to documents, or to the information inside them?

Consider, for example, two companies, both with field sales forces. The sales representatives at Company A may need the ability to access product datasheets from an online

library. A file-based system may suit their needs. But Company B's sales reps may need to retrieve custom profiles on any client; to satisfy this requirement, Company B would need to build unique documents on-the-fly from information components stored in a database.

Do You Want To Be Able To Reuse Document Information?

Information is incredibly valuable. Most organizations today realize that their information is more valuable than the applications used to capture it or the documents created with it. One of the forces driving interest in reusable document components are the many opportunities to "repackage" or "re-purpose" this information, especially for various types of electronic delivery.

What's the best method for managing information in a document-independent form? First, no matter where you put the information, it ought to be in a neutral format, such as SGML. This will ensure that information originating in a document created with a particular application can be reused in a document created with another application.

For applications built around reuse, file-based systems are often inefficient. While you can extract tagged information objects from document files and then recombine them to create new documents, doing this on a large scale is seldom efficient (just the time involved in opening files, much less searching them, becomes a factor). Stored in a neutral form as components (legal clauses, assembly instructions, maintenance tasks, encyclopedia articles, drug descriptions, etc.) in a database, however, information can be reused in virtually unlimited ways.

"The ideal database for documents would break them down into components, and recombine the components instantaneously in any way the user required."

Paragraphs about customer service, for example, could be reassembled and combined with other database information to create a unique brochure for key customers describing their own service plans. Other objects could also be reused in a marketing data sheet or a recruiting brochure. In the same way, an electrical specification created for a technical manual could be reused to create an RFP to subcontractors, or within a document specifying operating parameters for a diagnostic tool or flight simulator.

Component-level Document Management — Performance Issues

The ideal database for documents would break them down into components, and recombine the components instantaneously in any way the user required. We're not yet at the point where this is something you should expect to be able to do, although it can certainly be accomplished for some applications.

Document management at the component level can be very complex. The database has to keep track of a web of relationships that is far more complex than anything required in traditional transactions. It has to store the location of every component and the relationships between them as well as pointers to graphics and other external files. Documents often also contain cross-references to information components used in other documents.

Updating information across this web can be a massive task: While a traditional transaction will require one or two queries against the database, editing even a single paragraph in a document can spawn hundreds of database operations.

In addition to storing these physical locations, the database must maintain multiple logical versions of a document. There may be any number of revision levels and releases and configurations (variations in the way a document is assembled, or even specific information that is substituted, added, and deleted to match a particular customer's requirements). Sometimes these logical versions overlap (as in a particular release level of a particular configuration version), so that the same information elements are contained in different variations of the same document.

Of course, it's important to look at performance issues from the context of what you're trying to do with information. For example, if it takes a half an hour for a background batch process to reconstitute a document prior to final output maybe that's fine. Obviously, it's not acceptable if you have writers and editors waiting that long to retrieve documents they need to work on. In this case, you probably don't want to have to wait more than 3 to 5 seconds.

"... editing even a single paragraph in a document can spawn hundreds of database operations."

What Are Vendors Doing To Meet User Performance Expectations?

Nearly every vendor of document management solutions is working on improving the performance of its system. Several are taking very different approaches.

Xyvision's PDM, for example, maximizes performance by compressing document information and storing only what is different from one editing session to another. For each BLOB, there is a list that stores the beginning and end locations of changes made to each version of the document — which makes it possible to recreate the document as it was at any point in time. The same technique is used for storing SGML effectivity information (tags that indicate the parts of a document that need to change for different configurations).

IDI's BASISplus minimizes the time required to reassemble documents by storing components in a "prejoined and ordered" manner within a special "section record" object type. All BLOBs are indexed for full-text search and linked to tables that contain structural information.

Documentum says that in its next release, which will add the ability to store document components, the product will gain performance advantages from the way the Content Manager maps information attributes (stored in the relational database) to hierarchical content (stored in the relational database, in files, or on optical disk). An algorithm similar to a two-phase commit[2] will ensure that both attribute data and content data for a specific object are either committed or rolled back together. The same mechanism will be used to manage SGML effectivity.

Documentum will also improve performance by using intelligent caching to manage how document components and their attributes are communicated between the server and client.

Texcel expects some performance advantages to accrue as a direct result of the UniSQL database architecture. By combining object-oriented and relational capabilities in a single layer, UniSQL eliminates translations required in a layered solution. UniSQL also speeds navigation across relational tables by avoiding many of the join operations required in traditional relational architectures.

Also, because UniSQL allows one-to-many references, it can rapidly navigate a complex web of relations to manage document cross-references, for example, or information objects that are used in multiple documents.

The Future of Document Databases

We're moving rapidly beyond the era of "lights-out" database publishing. Increasingly we're asking databases to function not simply as storage media, but as tools that support people engaged in interactive and iterative work processes.

This has a lot of implications: We need more tightly linked tools that span the entire process from document creation, storage, revision, deliveries, and reuse. We need query

[2] The rule set by which a relational database either commits or rolls back related elements.

"... we're asking databases to function not simply as storage media, but as tools that support people engaged in interactive and iterative work processes."

languages that make it easy for end users and end-user applications to interact with the database. We need graphical, object-oriented tools that enable each set of users to create its own preferred views of the database.

In the same way that new WYSIWYG editing tools will eventually mask the complexity of adding SGML tags to documents, increasingly document-capable databases and document management solutions built around them will mask the complexity of structured, component-level document storage from the user.

The trends are pretty clear. Relational database vendors are adding object-oriented and document-savvy features to their products. OODB vendors will be keeping the pressure on as they work to standardize inter-object communication. Some key technologies that we will see more of in the future can be found today in commercial products and custom solutions. Examples include "active document" technology found in Interleaf and other publishing systems, document-oriented-interfaces to databases found in Texcel's use of the ArborText publisher as the interface to UniSQL, Document workflow GUI's like Xyvision's, and the combination of of relational, structural and full-text systems being built by InfoDesign, IDI, General Research, and Berger-Levreault.

As these kinds of capabilities grow more widespread in the next couple of years, it will become commonplace for electronic documents to be made up of content from different kinds of sources (See Figure 5).

Which vendors are likely to take the lead in offering these solutions? Well, clearly there are some early contenders, and we've described their approaches. It's also clear that they're going to face some stiff competition in the near future. Leading database vendors, for example, are beginning to build more sophisticated document management capabilities into their products — will they take the next step and provide their own document management solutions?

Risks and Costs

A major risk organizations face today is in not moving now to take control of their document information. The competitive advantages are so great — eliminating redundant work; facilitating information sharing internally and with suppliers, dealers, and other business partners; improving customer support; and creating new revenue-producing information products and services — that no company can really afford to wait.

At the same time, of course, it is important to choose your method of gaining control of your documents carefully. You need to ensure that your documents will be protected and reusable as databases and computing platforms change.

Be careful not to underestimate the complexity of the information in your documents. Your expectations will not be met if you think of documents as files when you need to manage graphics or part numbers that are buried in the file.

A significant cost in any database implementation is the data conversion required. If not carefully planned with reasonable expectations, the transition can be costly and disruptive. Other specific areas to watch out for when implementing a database-based document management system are: interoperability, performance, and data integrity.

It is difficult to cost justify a document management system that can't share information freely with document creation or distribution applications. Sluggish performance will either reduce productivity or encourage circumventing well-planned processes. Without sufficient checks and balances, cost savings can be wiped out by corrupt or inaccurate data.

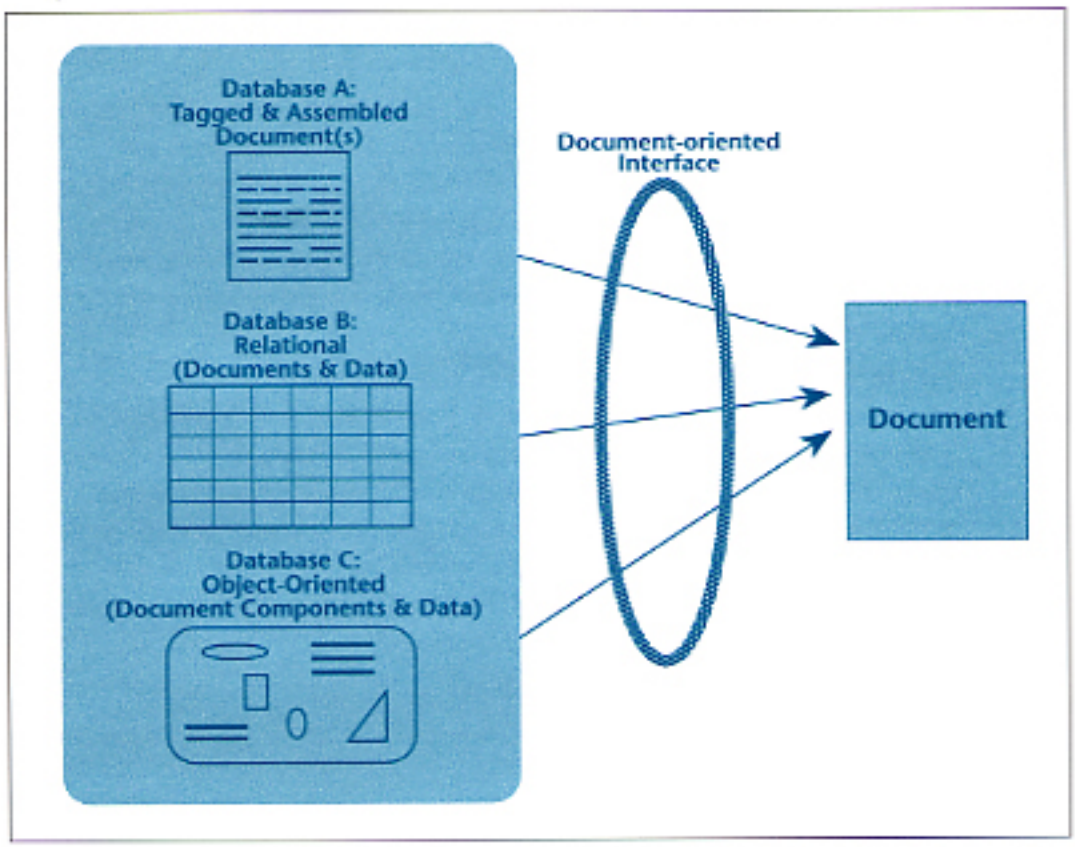

Figure 5

The Future of Document Management — Where Document & Data Management Meet

Another risk is choosing your document management solution too narrowly based on your current needs. In many cases, organizations first become involved with document management on a limited scale in a particular work group or department — but over time they begin to realize that this is an issue that impacts every area of their business. It is important to leverage document management solutions across the enterprise, while at the same time supporting the needs of local groups.

Conclusions & Recommendations

Many new document management products will come out over the next several years. But don't wait for them. Whether you're actually in line to make a purchase in the next 6-12 months or not, you should be laying the groundwork now.

1. Put the time in to analyze and understand your documents and workflow. The effectiveness of any database is largely determined by the quality of the data model. Think about who needs which documents when — but also consider who needs access to information *inside* the documents, and what they need to do with it. Each of the three types of document management products discussed makes different assumptions about those needs.

2. Consider the way you will be storing and accessing the information. Which file formats need to be supported? Which standards can help you reduce your risk? Explore the use of SGML for encoding your information (almost all the document management products covered provide some support for SGML).

SQL is the de facto standard for accessing information in databases, but you need to pick a version of SQL to use, and make sure your supplier's SQL implementation of it includes

the features you need. You also need to determine whether SQL by itself is sufficient for your document management needs (it won't be for everyone's).

3. Find the simplest way to support the kind of work you're doing now that does not lock you into a particular vendor's approach. For example, if you are already heavily committed to a particular database, consider one of the third-party products built around that database. In some cases, you can just "glue" them onto what you already have. Make sure that you feel confident, however, that the third-party product is designed in a way that will make it relatively easy for them to update it as the underlying database capabilities change.

If you're not heavily committed to a database, or if you're willing to support more than one, you might want to consider a product that uses a new hybrid (object-oriented and relational) database or a structured textbase. Depending on the type of information you need to store and use, the task of supporting a new approach may be more than compensated for by the performance gains and the ease with which you can manage your documents. If you do go outside of the mainstream, however, you want to be assured that the product can be integrated tightly with leading databases that future business partners might be using.

Rebecca Hansen

Topics Covered In Previous Issues

Vol. 1, No. 1.

What The Report Will Cover & Why — An Introduction To "Open Document Systems", And A Description Of The Report's Objectives.

Imaging, Document & Information Management Systems — What's The Difference, And How Do You Know What You Need?

Vol. 1, No. 2.

SGML Open — Why SGML And Why A Consortium?

Document Query Languages — Why Is It So Hard To Ask A Simple Question?

Topics To Be Covered In Future Issues

The subjects listed below are some of the areas we will be covering, in no particular order. If you have an opinion about which topics you would like to see added or covered sooner rather than later, let us know.

Electronic Distribution — Does One Size Fit All? Who Are The Players? What Are The Options? Are Pages Important?

Office Workflow Systems — Can They Handle Strategic Information, Or Are They For Casual Or *Ad Hoc* Use Only?

Documents As Interfaces — Is This An Option For Today? What Will The Future Bring?

SGML & Presentation Interchange — What Standards Are Available Or Appropriate? (DSSSL, OS/FOSI, HyTime, ODA, etc.)

Authoring Systems — Do You Need Different Kinds For Different Media?

"Middleware" — What Is This Layer Of Software In Between Operating Systems And Applications? Is It The New Proprietary Trap? What Does It Mean To Your Decisions About Document Systems?

ISO 9000 — What Kind Of Document Management System Do You Need To Meet This Quality System Standard?

Open Systems & Client Servers — What Are They? How Do They Relate To Document System Technology?

Document Elements & Distributed Objects — How Do They Relate To Each Other?

CALS & IETMS — What Are They? How Do They Influence Open System Technology?

Imaging Technology — How Is It Evolving?

The Airframe And Airline Industry's Strategy For Sharing Product Information — What Can You Learn From It?

New Drug Applications — What Document System Strategies Make Sense For The Pharmaceutical Industry?

Object & Relational Databases — Which Approach Is More Suited To Your Document Systems Needs?

Compound Document Architectures — Why Do We Need Them? Who Will Define Them? Will They Do What We Expect?

DOCUMATION '94 UPDATE

Because of the synergy between the Documation Conference and the topics covered in this report we will provide regular updates on the conference program and exposition.

Call For Papers Brochure Has Been Mailed

We have distributed thirty thousand preliminary brochures to interested parties in North America, Europe, and the Pacific Rim. Subscribers to this report who would like additional brochures should let us know how many you would like. A brochure with program details and registration information will be mailed in September.

Exhibitor Kits Are Available

Companies who would like to reserve space at the exposition should request exhibitor kits if they have not already received one. The 30,000 square feet in the exhibit hall is likely to be sold out well in advance. There are options for both small and large companies.

The Documation '94 Industry Advisory Board

Our Advisory Board Members are providing valuable assistance in formulating the content and format of the conference, they are:

Adobe Systems
Aetna Life and Casualty
American Honda
Andersen Consulting
Apple Computer
Avalanche Development Company
The Boeing Company
Computer Task Group
Documentum
EDS
Frame Technology
The Gartner Group
GEIE Gavel
InfoAccess
Intergraph Corporation
Interleaf, Inc.
McGraw-Hill
Merck & Company
Novell
Object Design
Oracle
R. R. Donnelley
SoftQuad
Sun Microsystems
Xerox Corporation
Xyvision

Further Information

To receive more information on the conference, or to receive an exhibitors kit, call Marion Elledge or Tanya Bosse at (703) 519-8160. If you have proposals for topics or speakers, call Frank Gilbane at (617) 643-8855, or fax your proposal to (617) 648-0678. Documation '94 will be held at the *Westin Century Plaza in Los Angeles, February 21-25, 1994.*

CALENDAR OF EVENTS

Below is a selection of key events covering open information and document system issues. There are many other conferences and shows covering related topics. We will attempt to keep this list to those events that focus on areas most directly related to the areas covered in our report.

Information & Technology Week. August 30-September 3, 1993, Anaheim, CA. GCA tutorials and seminar. Call (703) 519-8160, Fax (703) 548-2867.

CALS Europe '93. September 22-24, 1993, Berlin, Germany. Conference and exhibition on CALS-related activity in Europe. Call (202) 775-9556, Fax (202) 775-8122.

CALS Pacific '93. Fall 1993. Conference and exhibition for CALS activities in the Pacific Rim. Call (202) 775-9556, Fax (202) 775-8122.

Seybold San Francisco. October 20-23, 1993. San Francisco, CA. The enormous computer publishing exhibition and conference. Call (310) 457-8500, Fax (310) 457-8510.

CD-ROM Expo. October 27-29, 1993, Boston MA. Conference, tutorials, and exhibition. Call (617) 361-8000, Fax (617) 361-3389.

CALS Expo '93. November 1-4, 1993, Atlanta, GA. The annual conference and exhibition. Call (202) 775-1440, Fax (202) 775-1309.

Hypertext '93. November 14-18, 1993, Seattle, WA. Conference covering research in applications of hypertext-related technology. Call (212) 869-7440, Fax (212) 944-1318.

CALS Australia '93. November 17-18, 1993. Conference and exhibition for CALS activities in Australia. Call +61 3 819 6860, Fax +61 3 818 3129.

Explor. November 14-19, 1993, Denver, CO. The annual conference and exhibition on electronic printing systems. Call (310) 373-3633, Fax (310) 375-4240.

SGML '93. December 6-9, 1993, Boston, MA. The annual event in North America for SGML developers and enthusiasts. Call (703) 519-8160, Fax (703) 548-2867.

Documation '94. February 21-25, 1994, Los Angeles CA. The new annual international event for document management applications and document system technology. Call (703) 519-8160, Fax (703) 548-2867.

Order Form

❑ Please start my subscription to: The Gilbane Report on Open Information & Document Systems (6 issues). Back issues available for $45.

U.S.A.: $225 Canada: $232 Foreign: $242

Number of additional copies: ____. Add $35 for each copy mailed to same address.

Please send me additional information on:
❑ Consulting Services ❑ Special Reports
❑ On-site CALS Strategic Planning Seminar
❑ On-site Open Document System Strategic Planning Seminars

❑ My check for $ _____ is enclosed ❑ Please bill me

❑ Please charge my credit card ❑ MasterCard ❑ Visa ❑ American Express

Name as it appears on card _____ Number _____

Signature _____ Expiration date _____

Checks from Canada and elsewhere outside the U.S. should be made payable in U.S. dollars. Funds may be transferred directly to our bank: Baybank Boston NA, 175 Federal Street, Boston MA 02110, S.W. code BAYBUS33, into the account of Publishing Technology Management, Inc., number 1444-89-63. Please be sure to identify the name of the subscriber and the nature of the order if funds are transferred bank-to-bank.

Name _____ Title _____

Company _____ Department _____

Address _____

City _____ State _____ Zip _____ Country _____

Telephone _____ Fax _____ Email _____

Mail or fax this form to: Publishing Technology Management, Inc., 46 Lewis Avenue, Arlington MA 02174-3206
Fax: (617) 648-0678 • To order by phone call: (617) 643-8855

How To Find Out More About Companies Mentioned In This Issue

ArborText, Inc.
1000 Victors Way, Suite 400
Ann Arbor, MI 48108
(313) 996-3566

Berger-Levrault/AIS
34, avenue du Roule
Neuilly-sur-Seine, 92200
France
+33 1 46 40 10 60
Fax +33 1 46 40 18 44

Digital Equipment Co.
200 Forest Street
Marlboro, MA 01752

Documentum
5724 West Las Positas Blvd.
Pleasanton, CA 94588
(510) 460-4115

Fulcrum
785 Carling Avenue
Ottawa, ON K1S 5H4
Canada
(613) 238 1761

General Research Corporation
1900 Gallows Road
Vienna, VA 22182
(703) 506-5310

InfoDesign
100 Wellesley St. East,
Suite 100
Toronto, ON M4Y 1H5
Canada
(416) 928-6800

Interleaf
Prospect Place, 9 Hillside Ave.
Waltham, MA 02154
(617) 290-0710

Information Dimensions, Inc.
655 Metro Place South
Dublin, OH 43017-1396
(800) 328-2648

Object Design
1 New England Executive Park
Burlington, MA 01803
(617) 270-9797

OfficeSmiths
Division of CTMG
11 Holland Ave. Suite 700
Ottawa, Ontario
Canada
(613) 729-2043

Ontos Inc.
Three Burlington Woods
Burlington, MA 01803
(617) 272-7110

Odesta Corporation
4084 Commercial Avenue
Northbrook, IL 60062
(708) 498-5615

Open Text Corporation
Suite 550,
180 King Street South
Waterloo, ON N2J 1P8
Canada
(519) 571-7111

Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
(415) 506-7000

Sybase
6475 Christie Avenue
Emeryville, CA 94608
(510) 596-3500

Texcel
Texcel House
65A Alma Road
Windsor, Berkshire, SL4 3HH
United Kingdom
+44 753 810 678
Fax +44 753 810 680

UniSQL, Inc.
9390 Research II, Suite 200
Austin, TX 78759
(512) 343-7297

Verity
1550 Plymouth
Mountain View, CA 94043
(415) 960-7600

Workgroup Solutions
76 Blanchard Rd.
Burlington, MA 01803
(617) 229-9000

Xyvision, Inc.
101 Edgewater Dr.
Wakefield, MA 01880
(617) 245-4100

© 1993 Publishing Technology Management, Inc. All rights reserved. No material in this publication may be reproduced without written permission. To request reprints or permission to distribute call 617 643-8855.

The Gilbane Report and PTM are registered trademarks of Publishing Technology Management, Inc. Product, technology and service names are trademarks or service marks of their respective owners.

The Gilbane Report on Open Information & Document Systems is published 6 times a year.

The Gilbane Report is an independent publication offering objective analysis of technology and business issues. The report does not provide advertising, product reviews, testing or vendor recommendations. We do discuss particular pieces of product technology that are appropriate to the topic under analysis, and welcome product information and input from vendors.

Letters to the editor are encouraged and will be answered. Mail to Editor, The Gilbane Report, Publishing Technology Management, Inc., 46 Lewis Avenue, Arlington, MA 02174-3206, or fgilbane@world.std.com or APPLELINK:PTM

ISSN 1067-8719