FERRIS MESSAGING ANALYZER

analysis and practical insight for messaging professionals

May, 1995 Volume 4, Number 5

DBMS In Message Stores & Directories

Summary

Database technology is beginning to have a big impact on message stores and directories, and in this *Analyzer* we discuss why. We first explain what a database management system (DBMS) is.

Then we explain the big benefits for a message store: relatability, searching and sorting, multiple object types, backup and recovery, etc. We discuss the relative merits of a *closed* versus an *open* DBMS. Turning to directories, we discuss how a DBMS helps, and the main user requirements: hierarchical support, flexible and efficient searches, extensible data structures, security, and so on. We conclude with a discussion of the future of message stores and directories from the DBMS standpoint.

David Ferris
mom.

In This Issue

Who Should Read This?

Messaging professionals, IS management, database professionals, applications development staff.

Pick of the News

Tandem announced a range ctronic commerce solutions: low- to high-end Web servers, strong encryption including hardware encryption, a way of providing for electronic payment, plus consulting and implementation services.

1111111111111111111111

Back Issues

Recent back issues are: Supporting the Traveling User (4/95), Assessment of HP OpenMail (3/95), Definition of a Messaging System (2/95), Organizing E-Mail Support (1/95), Messaging Support Tasks (12/94), Messaging Management Standards (11/94), Assessing Lotus/SoftSwitch EMX/LMS (10/94), Ten Critical Requirements for LAN E-Mail Scalability (9/94), Forms Workflow Applications (8/94), Public Key Cryptography (7/94), Cross-System Directory Synchronization (6/94), MAPI & the X.APIs (5/94), CMC, VIM, & Other Messaging APIs (4/94), Directory Propagation (3/94), Performance Management Tools (2/94), Ensuring Timely Delivery (1/94), Stress Testing X.400 Gateways (12/93), File Transfer (11/93).For full details about these and other back issues, contact us at:

Tel: +1 415 986 1414 Fax: +1 415 986 5994; Ferris@CIS.Compuserve.com

111111111111111111

Ferris Messaging Analyzer is published by Ferris Research, 353 Sacramento Street #600, San Francisco, CA94111. Tel (415) 986-1414. $495/year, single copies $75. Reprints and consulting also available. The title *Ferris Messaging Analyzer is* a trademark of Ferris Research. Nothing that appears in *Ferris Messaging Analyzer* may be reproduced in whole or in part without the permission of Ferris Research. © 1995 Ferris Research, Inc. All rights reserved. The material contained in *Ferns Messaging Analyzer* is based on information Ferris Research believes is reliable, but its accuracy and completeness cannot be guaranteed. No liability is assumed for the use of any materials presented in *Ferris Messaging Analyzer,* nor for any errors or omissions which may remain. Contact Editor David Ferris at (415) 986-1414 or X.400: G=david; S=ferris; A=mci; C=us; DDA=ID=3773271 or Internet: 75300.3422@compuserve.com. Copy Editor: Michael Miley. Page Layout: Kristina Jaffna. For sales information, call or e-mail us at Ferris0 CIS.0 omp us erve.com.

What a DBMS Is

A database management system is a set of software tools that help you store data, keep it updated, and generally use the data. The topic's a deep one, but here is a quick summary of the main things a DBMS offers:

Data Definition

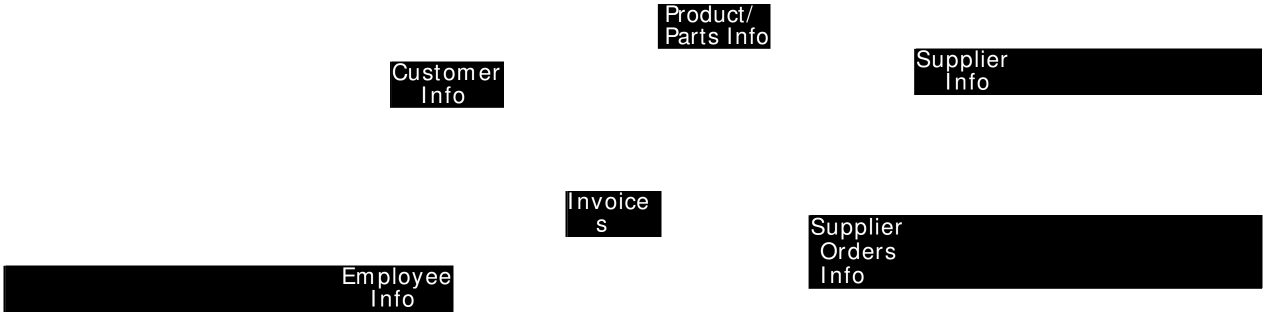
A DBMS lets you define data structures corresponding to business needs. Eg, you could define a human resources database, or an oil test sample database, or an accounting database. Once this has been done, data can be entered into the empty database, applications can be written to access it, and so on.

Data Relatability

When you start to model a business process, certain information is naturally clustered together; eg, someone's name and their social security number, or a customer's name and their ID. Conversely, some information seems naturally to be kept separate, even though it's related. So as databases grow, they usually become populated with different types of information, with one type of information able to point in some way to other types of information.

Figure 1 shows the sort of clusters of information you might see in a manufacturing firm. Here, invoices are sent to customers, and invoices related to products made by the firm. Likewise, the firm keeps track of orders it has placed with suppliers. There are lots of different types of interrelations here; eg, many parts can be supplied by a series of competing suppliers, many parts are made up of subcomponent parts, each invoice can be associated with one or more sales staff, and each sales rep has a number of associated customer accounts to maintain.

FIGURE 1 DATA RELATABILITY



An example of some of the clusters of information you might see in a manufacturing firm, together with an indication of the relations between the data.

It can get pretty complex, and designing a database to model business processes is something of a black art. But the point here is: a DBMS provides a way of modelling different types of data, and of defining interrelations between the different types of data.

Note in this *Analyzer,* the term "relatability" refers to the ability to associate different types of information. We are not referring to a specific DMBS technology, that of the relational DBMS (eg Oracle, Sybase); in our sense, both relational and non-relational DBMSs can define relations between data.

Shared Resource

As figure 1 illustrates, a DBMS is typically used to store many different types of data. Many different applications programs and many different user departments are likely to access the data. So a DBMS lets the data be accessed as a shared resource.

Integrated 4GL

A DBMS allows programs written in high level languages like C and Cobol to access the data. These languages provide a high level of control, but they take a lot of time to develop and maintain. Thus a DBMS normally has a still higher level procedural language which simplifies applications development—often called a *4GL* or *fourth generation language.*

Ad Hoc User Access Utilities

Users will often access data using applications that have been written by a programmer, probably in a 4GL. However, they also need to be able to access data when a program hasn't been specially written. DBMSs normally provide an ad hoc data query and update tool, and a report writer.

Fast Access

Many people, perhaps in the hundreds, may need to access the same information at the same time. A DBMS typically has a lot of intelligence built in to let this happen with acceptable response times.

Concurrent Access

Several people may try to access the same information at the same time, such as a message on an airplane seat reservation, a bulletin board, or an electronic calendar. Problems arise if at least one person can update the information; a DBMS provides special facilities to ensure that everyone gets an accurate, concurrent view of the data.

Access by Third Party Tools

Notwithstanding the availability of built-in application development tools, a DBMS will provide a way of getting at the data using third party tools such as PowerBuilder. Often, communication is via Microsoft's *ODBC* (Open Database Connectivity), a standard data access API.

Access Security

A DBMS lets you restrict access to information. Options will include read-only, update, and no-access privileges, and this will probably work down to the individual field level. This stops unauthorized users from getting at sensitive information such as someone's salary, or updating information they shouldn't (eg, most users shouldn't be able to increase a salary field, even if they have read-access to it).

Backup & Recovery

Commercial DBMSs offer a means for continuously backing up the data, so that if there's a fault, the database can be reinstated automatically, quickly, and accurately. The database is available to users, even when backups are taking place.

Message Store as a Database

Older e-mail systems typically use directories and files to store messages. Often, each user has a directory, and individual text messages and file attachments are each stored as individual files. NetWare Global MHS is the best-known such system, used by such products as ON Technology's eMAIL (ex-Da Vinci) and Notework (figure 4 on page 16).

DBMS Benefits

However, over the last few years, most messaging vendors have incorporated some sort of DBMS, and they use this to define and maintain the message store.

The big advantages, as compared to using a file system, are:

Concurrent Access & Relatability

The built-in support for concurrent access, and associations between different pieces of information, makes it much easier to provide access to shared objects, such as stock price information in a bulletin board, or a calendar database.

The support for pointers to associated information also makes efficient attachment handling a lot simpler:

- It reduces the number of copies that must be made, eg, when the same file is being sent to a number of users. This by itself typically cuts the size of a message store by some 50% to 75%.
- It reduces the amount of traffic that must be transferred over slow WAN links. In some cases, a pointer rather than the file can be transferred, so that actual transfer need only take place if the file is needed. Lotus Notes is a good illustration: you can click on an icon and be transferred to that document, even though it may reside in a separate store.

Searching & Sorting

A DBMS has a lot of retrieval facilities—indices, hash tables, B-trees, etc. This makes searching and sorting on various criteria—subject, author, creation date, etc—much easier and faster.

Different Types of Structured Object

Message stores contain more and more types of special objects, together with programs that know how to process these objects. Eg, in addition to a directory, text messages, and attachments, you may well find voice messages, fax images, electronic forms, electronic calendars, and bulletin boards. A DBMS makes it much easier to define and manage these objects.

Scalability

Many messaging objects have components which range enormously in size. Eg, a file attachment can range from a few bytes to many megabytes. A DBMS can handle such variations more easily than the DOS file system.

Security

Since message stores contain more and more shared information, access controls are needed, often to sub-elements of a message. Eg, *no-access, read-only access,* and *update-access* privileges may be needed for a field in an electronic form. A DBMS provides the necessary machinery.

Replication & Synchronization

Message stores must provide for the automatic copying of certain types of information between different message stores. Eg it has to transfer directory update information, bulletin board entries, and calendar free and busy times.

Several things can make this quite tricky:

- Often, the system must be able to check what changes have been made to similar objects, and then consolidate changes in some way.
- Sometimes links are only available intermittently, as is the case with a dialup user who needs his/her directory to be updated.
- The criteria for what is transferred can vary dynamically, as with a dialup user who requests that only messages under 50KB in size should be synchronized between the office and laptop message stores.
- The information may be transferred as mail messages, or using interactive communications.

With a DBMS, a set of tools can be built by the messaging vendor that provide for replication and synchronization, and these tools can easily be made to work with all the different DBMS objects.

24 by 7 Operations

A DBMS's backup and space management utilities are unlikely to require the system to go off-line.

Most Messaging DBMS are Closed

Thus, DMBS technology has much to offer a messaging system. Usually, the messaging vendor builds its own DBMS system and tools, and then implements its message store using the DBMS.

The quality of these proprietary DBMS varies enormously. Some, such as that of cc:Mail, offer very rudimentary services. MS Mail is stronger in this regard: eg, supported objects include meeting requests and electronic forms; there's automatic data compaction and space reclamation; and administrators can add new fields to the directory.

Novell's *GroupWise* (v4.0 onwards) has perhaps the best built-in DBMS. Eg, message store compaction takes place dynamically, without bringing the system offline; administrators can add new fields; the message store supports many different types of interrelated objects; message store objects are accessed and managed using consistent tools; and directory replication is simple to understand and reliable.

GroupWise also takes advantage of the fact that users do not have general-purpose access to the DBMS. As a result:

- The DBMS is relatively compact and can be run even on a low-end 386 client.
- It has been ported to many different platforms—notably Windows, DOS, Macintosh, UNIX, 0S2, and NLM environments.
- The code has been written to run in user workstations or on servers, depending on customer requirements.
- The cost of the code is lower than it would have been had a third party's DMBS been used.
- Revision synchronization problems across platforms are reduced, since Novell controls the code.
- Performance is optimized for the particular database structures used.
- The functionality needed specifically for a message store could be given priority.

Advantages of an Open DBMS

These sort of consideration make it likely that most DBMSs used for message stores will continue to be closed, proprietary systems for the foreseeable future.

Nevertheless, there are some considerable advantages to using an *open* DBMS. This is where you can pick and choose a commercial DBMS to use for the message store, and where you have full access to all the services provided by the DBMS. In particular, an open DBMS is likely to offer:

- A 4GL to process message store entries.
- Access to data controlled by other departments. Eg, the messaging system could access databases controlled by the personnel department.
- Ad hoc query/update/report writing tools.
- Good support for third party tools using an ODBC interface.
- Easier object extensibility. Eg, if you want to add a couple of new fields to the directory, you can.

- Far better backup and recovery services.
- Easier access by external production applications.
- Better access by third party development tools.
- Better space-management.

Future Directions

We finish this section on message stores with a few comments about what to expect during the next five years.

Users Living in their Message Store

Users will increasingly "live" in the message store. It will be the main thing they look at, literally, when they are not ensconced in a particular application. It will be the main way they become aware of communications from colleagues.

Mostly Closed DBMS

As we noted, it seems likely that the DBMSs used to implement message stores will remain closed for the next few years. Access by external programs will be somewhat unwieldy, mostly by means of MAPI interfaces and import/export utilities.

Separation of Applications from Message Store

There will be an ever-cleaner separation of the message store from the applications that use it, by means of an API. Microsoft's MAPI is a clear example. Recently, Oracle demonstrated a MAPI driver working with the Microsoft Exchange In-Box, and Lotus demonstrated its client software working with MAPI back ends. Lotus' VIM also illustrates an application-to-message-store API, and ISODE Consortium is working on such an API for its X.400 software.

In theory, these APIs should allow users to insert their message store of choice. As a practical matter, however, most of them focus on message store access, and only allow very limited management control. Other utilities provided by the messaging vendor must be used to do things like add new fields, run garbage collection, do backup and recovery, and so on. These management tools work only against the vendor's DBMS.

Ever-Richer Message Stores

Vendors will continue to enhance message stores, adding in more and more special object types, as is the case with GroupWise. Users will also be granted greater flexibility to introduce objects of their own choosing.

Support by Mainstream DBMS Vendors

Some messaging products—notably message switches—use third party DBMSs. Two that come to mind are *AlisaMail,* which uses Sybase; and Lotus' *LMS* (ex-EMX), which uses Oracle.

However, DBMS vendors have yet to build in facilities that specifically try to accommodate a messaging system. Eg, they could provide edit masks that define SMTP or X.400 e-mail addresses, or they could build a MAPI service provider. Given the advantages of using an open DBMS to support a message store, it seems likely some vendors will take steps in this direction. Oracle, in particular, is probably worth watching.

(continued on page 16)

Directories As Database

We turn now to discussing the role that DBMSs have to play in implementing directories.

The DBMS capabilities of most messaging system directories are extremely crude today. Eg:

- There are a very limited number of fields—often just address and comments fields; or address, name, and comments fields.
- Administrators cannot add additional fields.
- You can't access the directory using SQL commands.

Nevertheless, exceptions exist. *Oracle Office* has probably the most sophisticated DBMS capabilities found in a messaging product today. No prizes for guessing what DBMS its directory uses. As implied earlier, Novell's GroupWise also has a powerful DBMS-based directory.

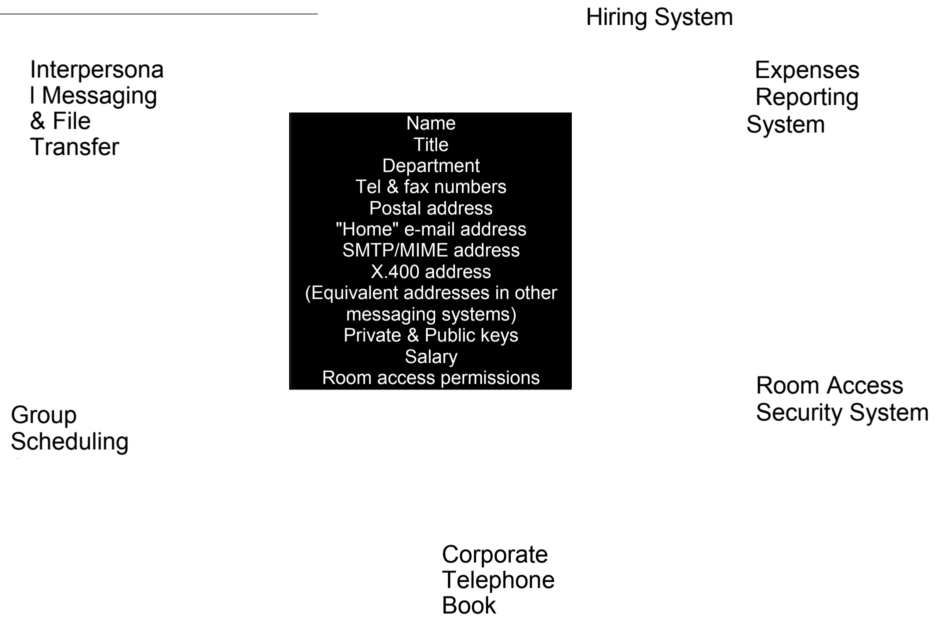
More Complex Data Modeling

Messaging system directories are being upgraded because the demands on them are becoming a lot more complex. Eg, vendors find that maintaining the replication machinery is a big headache, and they'd like to find a more sophisticated infrastructure.

However, the main reasons have to do with the growing richness of directory information, and the recognition that the information needed by a messaging system is very closely related to other types of corporate information (figure 2):

- Messaging systems deal with ever more complex objects (see preceding discussions of message store). As they do, the directory must be expanded. Eg, if a calendaring system is included, knowledge about access privileges must be provided so that secretaries can access their boss's diary; and information about people's roles and public and private keys must be provided for forms routing applications.
- It's often natural to add non-messaging information to the messaging directory. Eg, many organizations want to add their corporate telephone book to their messaging system directory. Using familiar e-mail utilities, users can then easily look up someone's telephone number, title, department, mailing address, fax number, and so on.
- Conversely, it makes sense for many organizations to have the messaging system access databases owned by other departments. The most obvious con-

FIGURE 2 TOMORROW'S DIRECTORY



The information needed by a messaging system—and stored in a directory—is becoming much more complex, and is being used by a growing portfolio of applications.

tender is to let the messaging system access the database owned by the Human Resources department. This is likely to contain such personal information as: employee ID, name, title, salary, department, social security ID, telephone and fax number, postal address, e-mail address, and physical access clearances (for use in security card systems).

In sum: the messaging system directory is no longer a simple, specialized, isolated system. It now participates in the tangled web of corporate information used by many applications and many departments.

DBMS Requirements

Many of the technical demands of this new environment are satisfied by migrating the directory to a DBMS. The most important of these (figure 3) are:

User Requirements

- Hierarchy Support. It should be possible to represent data hierarchically. Eg, a primary use of an enterprise directory is to find out information about employees. To aid users in browsing, the directory of a large enterprise should usually have a hierarchical structure matching that of the enterprise.
- Flexible Searches. The most common way to use a directory is to search for entries. The ideal directory will support the ability to search not only on name attributes—eg, find all the entries with a name attribute value of *Charles Smith*—but also other attributes—eg, find all entries with mail stop *MS-1402*. So the search capability should be flexible.

FIGURE 3 KEY DIRECTORY REQUIREMENTS PROVIDED BY A DBMS

User Requirements	**Technical Support Requirements**
Hierarchy Support	Access to Other Data Sources
Flexible Searches	Easy Change Management
Efficient Searches	Extensible Data Structures
Replication	Large Directories
Multiple Platforms	Centralized vs. Distributed Administration
	High Volume
	Access Security
	Choice of Directory Server Platforms

Migration to a DBMS infrastructure makes a directory much more valuable for users and technical support staff alike.

- Efficient Searches. Especially when searching the directory, users expect to get the entry information they want quickly. As a guideline, queries like "find all entries with last name starting with *macd*" should be resolved within a second.
- Replication. This helps improve response time and information availability.
- Multiple Platforms. The directory should be available on various platforms, such as DOS, Windows, Macintosh, UNIX, and 0S2.

Technical Support Requirements

- Access to Other Data Sources. There should be mechanisms for accessing data from other sources, such as HR databases or the security control system; eg, to facilitate initial population of the directory.
- Easy Change Management. Modern organizations are typically in a state of constant flux, and entries are changed often. Administration tools should make modifying entries easy.
- Extensible Data Structures. It should be easy to add new fields and structures to the directory.
- Large. Directories. The directory should be able to grow to large numbers of entries with acceptable performance.
- Centralized vs Distributed Administration. An organization should be able to choose how it wants to administer data. Eg, a highly centralized HR department may need centralized administration; conversely a distributed HR department (as is the case with supermarket chains, say) may need to integrate many separate databases.
- High Volume Access. Other systems may need to extract large volumes of information quickly.
- Security. The sensitivity of information contained in a directory grows with time. Consequently, it is necessary to restrict access to some of it. This requires the ability to authenticate the identity of clients seeking access and the ability to control the type of access.

- Choice of Directory Server Platforms. If at all possible, directory server software should run on various platforms, so that customers can run it on computers they already own and with which they are familiar.

For a more detailed discussion, see *Implementing Corporate Directories,* published by Ferris Research (call +1 415 986 1414 for details).

Future Directions

We end with some comments on changes messaging directories are likely to experience as far as DBMS technology is concerned.

Migration to DBMS

A we've noted, most messaging vendors are only just beginning their migration to DBMS technology. Expect big improvements in functionality and manageability over the next few years.

Closed Technology

In principal, it makes enormous sense to let a messaging system access an external directory, because of the growing need for cross-application directories. User organizations would like to manage this resource using their standard DBMS technology—ie, using products like Oracle and Sybase.

Nevertheless, it appears that in the immediate future, most directories in messaging systems will use their own DBMS:

- The cost of a third party DBMS integration will discourage its use given the market price per seat.
- Performance can be optimized to meet directory needs.
- Developing the functionality required—which is way short of a full-blown commercial DBMS—is not impractical.
- DBMS vendors offer limited support, if any, for the special needs of messaging directories.

In cases where the messaging vendor incorporates a third party product, you still should not expect to have general access to the underlying DBMS. The messaging vendor will have defined special data structures, and is unlikely to want these to be seen, let alone altered.

X.500 Based on DBMS

In theory, X.500 directories can be stored in commercially available DBMSs. Lotus illustrates: it's using Oracle as the basis for its offering. However, when this occurs, don't expect the underlying database to be be accessible by other applications. It will be full of X.500-specific support stuff, like special pointers.

API Access

Over the short term, API access will be mainly via MAPI directory service providers, and import/export utilities. The X.API's *XOM* and *XDS* APIs won't be used much, because they are too hard to learn. As with message stores, SQL will gradually become a popular way to access directory information, although its use is rare today.

NOS Integration

We have already noted the overalap between messaging directories and Human Resources databases. However, they also have a strong overlap with Network Operating System (NOS) directories, and there will be increasing integration between the two. Early players here are Banyan's *StreetTalk,* with excellent integration; Novell, too, is working hard to integrate NetWare's *NDS* directory with messaging directories.

Microsoft's newly announced *Open Directory Service Interface* strategy is also worth following. It may end up defining standard NOS-level calls to access a variety of different directories. However, the initiative is in an early stage of development and its success is far from established.

Object-Oriented DBMS

Most DBMSs aren't object-oriented, whereas many messaging directories are already committed to this direction. Clearly, underlying DBMSs will, over the long term, need to offer this capability.

Lotus European Technology Conference, April 1995

Over 1,000 people attended Lotus's European Technology Conference in late April at Brighton, in the UK. Attendees were mostly e-mail administrators, database managers, and strategic planners from large organizations.

Lotus revealed its plans for the future, highlighting the communications side of the company. The company is gambling heavily on the success of Notes, and this conference showed it.

Although *WordPro,* the revamped version *of AmiPro* was announced, the majority of product announcements and demonstrations described new products and enhancements for Notes. These included Internet publishing and news gathering, Notes administration, and a demonstration of the forthcoming cc:Mail post office and client. The biggest crowd puller was a preview of Notes release 4, which among many other enhancements, includes an e-mail user interface that looks like the cc:Mail client.

The Internet played a large part in the conference. In addition to Web publishing, cc:Web was previewed. This will allow Internet access via a cc:Mail front end, and give access to cc:Mail post offices via the Internet.

60 Business Partners showed a variety of third party products and services, including a new pager product from C2C, new e-mail gateways from Viatec, and 28.8Kbit/s access to Worldcom's Public Notes Network.

cc:Mail's Future

The future of cc:Mail was shown to be a complicated migration to Notes.

The first step will be a new client, version 6, adding no real client functionality, but necessary to access the next release of the cc:Mail database, version 8. There will also be new releases of the cc:Mail MTA (Router 6) and its directory updating and synchronising product, ADE (Automatic Directory Exchange). This release of the database will allow on-line reclamation and repair of the central database, a feature long awaited by mail administrators. At present they have to wait until no one is using e-mail to do these tasks.

After mail administrators have upgraded all users, the databases, and all the administration tools, they will then be able to upgrade the users all over again, to the next level of the client, version 7. This will allow access to Notes release 4 mail systems via a link product and rich text messages. After this, cc:Mail will be supported, but it is likely that no further versions of the mail engine will be developed, Notes being preferred.

David Powell

Research Method, Vendor Contacts

We interviewed a number of vendor technical managers, threw in our own views, stirred them together for a week or two, and this is the result. Our thanks are due to Bob Denny (WVVVV software developer and ex-CEO of Alisa Systems), David Goodman (Lotus), Eldon Greenwood (Novell GroupWise Division), Steve Kille (ISODE Consortium), and Mark Stieglitz (Lotus). We also drew upon material in *Implementing Corporate Directories,* a research report and information service from Ferris Research. Finally, Microsoft's Dilsad Uvez read the text for errors and omissions and gave us several valuable suggestions.

As usual, because company policy often precludes public comment by users on specific products, and because interviewees need to be able to speak freely, the views expressed are presented without attribution.

Development Team

David Ferris writes and edits the *Ferris Messagin g Analyzer* He edited and co-authored three research reports, *Implementing X400 Backbones, Implementing Corporate Directories,* and *Integration of PC E-Mail: Planning, Product Evaluation, Implementation,* published by Ferris Research. Thanks to e-mail, fax, and cheap phone lines he lives both in San Francisco and London and works on a day-to-day basis with colleagues in several countries. He has an M.S. in Computer Science from Stanford University, California.

David Powell, who wrote the review of the Lotus European Technology Conference, is a freelance e-mail consultant working in the UK. He started his career in corporate planning and strategic marketing, and became IS Manager for British before going solo in 1993. He was one of the founders of the European cc:Mail User Group, and has edited, written, and published its magazine since then.

April News

VARIOUS ... Document Management Alliance. Task force of *Association for Information and Image Management* will develop industry specs for document management interoperability; merges DEN and Shamrock initiatives. Jason Werner, Novell, +1 801 228 5144. ***JetForm & PeopleSoft.*** JetForm will integrate its forms software with PeopleSoft financials, distribution, and human resources apps. Mark Demers, JetForm, +1 703 934 6183. ***Memo for Windows v2.1, Memo/Base v5.1.*** Attachment handling, cc/bcc: recipients, directory views, client management & notify, TCP/IP support, multi open mailboxes. Adam Sroczynski, Verimation, +1 201 391 2888. ***Oracle Office PROFS/OV Gateway, Windows SPL*** Gateway includes calendaring integration; Windows SPI in development for Oracle Server. Steve D'Alencon, Oracle, +1 415 506 9454. ***Hitachi Directory Synchronization for Notes, Group Wise.*** *Agent@Notes & Agent@GroupWise* gateways integrate with other directories including those of cc:Mail, MS Mail, PROFS/OV, QuickMail, Sendmail. Rick Davis, Hitachi, +1 408 986 9770. ***EDI/EC Interface for HP OpenMail.*** Frontec AMT introduced an interface between its EDI & electronic commerce software. Vic Werner, Frontec, +1 203 977 7107. ***ON Technology's Common Knowledge.*** Collaboration groupware product for sharing information and documents now shipping. Steven Guthrie, ON Technology, +1 617 692 3106. ***Visioneer's PaperPort for Workgroups.*** Document images now sent more easily by popular LAN e-mail packages. Eric Zetlin, Visioneer, +1 415 812 6483. ***Commercial Wireless Data Service.*** McCaw launched its *AirData Business Service* for mobile workers, uses CDPD. Beth Bousley, McCaw, +1 206 803 4050. ***SMARTslip for Macintosh.*** ChorusSoft shipped SMARTslip, an electronic routing slip for documents. R.C. Venkatraman, ChorusSoft, +1 408 736 4024. ***Discussion's Conferencing.*** Information sharing software for MS Mail, cc:Mail, GroupWise, planned for June-Fall 1995 release. Srivats Sampath, Discussions, +1 503 690 1929. ***Using Email Effectively.*** $14.95 book contains practical tips on basic e-mail use. Sara Winge, O'Reilly, +1 707 829 0515.

LOTUS ... ***Desktop &Mobile cc:Mail v2.2.*** More than 150 new & updated viewers, easier composing of foreign addresses, now available. ***Technical Support Via World Wide Web.*** Lot of good information free at http://www.lotus.com. ***Notes & cc:Mail Imaging.*** *Lotus Image Viewer* and a print-to-fax driver to be bundled free. Other elements add annotation and rich scanning, support for optical disk drives & jukeboxes, OCR, developer interfaces, available May 95. ***Lotus Fax Server.*** For cc:Mail & Notes, available May 95. ***Phone Notes Application Kit v2.0.*** New forms, more flexible interface, better manipulation of playback on voice & text objects, now shipping. ***ScreenCam for Windows v2.0.*** Captioning, sound compression, editing & Notes/FX integration, now available. Elena Fernandez, Lotus, +1 617 693 1606.

FAX ... ***EMA & International Computer Facsimile Association Merge.*** ICFA will form the *EMA's Computer Facsimile Committee.* Alison Graham, EMA, +1 703 524 5550x227. ***Delrina's CommSuite for Networks.*** Shared access to LAN fax servers, network access to Delrina's *Fax Broadcast* service, available June. ***Delrina's WinF ax PRO 40 CD.*** Includes listing of over 100,000 published business fax numbers, 2,000 click art images, over 200 humorous cover pages. Shelly Sofer, Delrina, +1 416 441 4702. ***Voice & F ax Development Tools.*** *Show N Tel* uses Brooktrout boards. Raymond

Phillips, Technically Speaking, +1 508 881 1900. *Brooktrout/FaxBack Litigation.* Settlement grants FaxBack a license to use fax-on-demand method described in Brooktrout patent. Heather Magliozzi, Brooktrout, +1 617 449 4100 x263.

INTERNET/SMTP/MIME ... *Tandem's CyberWeb.* Family of electronic commerce solutions includes multiprocessor Web server, S-HTTP and SSL security, payment by connection to Open Market's *Integrated Commerce Service,* hardware-based cryptography, consulting. Available summer 1995. Lisa Copass, Tandem, +1 415 274 8119. *PSI's Pipeline USA. New* internet connection service. $19.95/month for unlimited up-to-28.8Kbit/s dialup. 120 US POPs planned for mid-1995. Jeff Luther, +1 703 810 1517. *HP OpenMail Gets MIME, UUENCODE/DECODE.* Enhanced gateway/transport, now available. Jeannette Tollstrup, HP, +1 408 447 5333. *WWW Server for Macintosh.* StarNine will sell *WebSTAR* and *WebSTAR Pro* in addition to other Internet software for Mac servers, shipping by June. David Thompson, StarNine, +1 510 649 4949. *SecureWeb Toolkit.* Lets WWW developers produce exportable secure Web browsers and servers. Barbara Kline, Terisa Systems, +1 415 694 7991. *Vermeer's Pavilion.VVVVVV* publishing tool will ship in July. Brenda Nichols, Vermeer, +1 508 369 2100.

X.400/X.500 *NEXOR's 1993 X.500. Message Ware Directory Server* has better performance, management, standard access controls, replication, directory synchronization API; no longer based on QUIPU. Nick Laszlo, NEXOR, +44 115 952 0506. *Supply Tech Offers X400 RUA.* EDI firm will resell MaXware's X.400 RUA. +1 313 998 4800.

FINANCE & BUSINESS ... *Lotus Results.* Quarter ending April 1 had $17.5 million net loss on $203 million revenues. Elena Fernandez, Lotus, +1 617 693 1606. *FirstClass Market Share.* SoftArc claims 3 million licenses, says is about to become 3rd most popular PC e-mail package after cc:Mail and MS Mail. Dallas Kachan, SoftArc, +1 905 415 7000. *Lotus' Campus-Wide Licensing.* Small costs for everyone in a university. Elena Fernandez, Lotus, +1 617 693 1606. *Compuserve Size.* CompuServe claimed 3 million active accounts, 60,000 new users per week. Russ Robinson, CompuServe, +1 614 538 4274. *CompuServe W* Viewer will provide frequently refreshed still video images & real-time closed-captioned text, available late 1995. Carrie Reber, CompuServe, +1 614 538 4092. *XcelleNet & McCaw.* Strategic alliance will integrate McCaw's *AirData* network, XcelleNet's *RemoteWare.* Terrie O'Hanlon, XcelleNet, +1 404 698 8650 x111. *Brooktrout Results.* 1Q95 had $6.7 million revenues, net income $710,000. Bob Leahy, Brooktrout, +1 617 449 4100 x226.

I■111111

Message Store Future Directions

(Continued from page 8)

Object-Oriented DBMS

There will be growing interest in the used of object-oriented message stores. Many important messaging models are already object-oriented, notably MAPI, the X.APIs, and X.400 and X.500 systems. Message stores will need to follow in their footsteps.

Free-Form vs Structured Objects

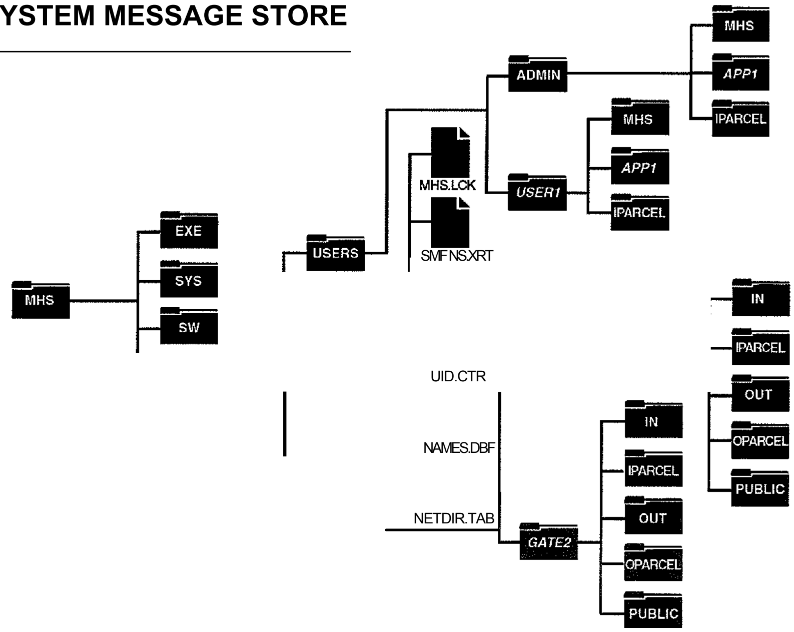
Some message stores—notably that of Lotus Notes—allow users to enter a very wide variety of unstructured documents. Other message stores usually require very well-defined objects. There's a direct analogy here between the sort of objects than can be defined in the World Wide Web, and those than can be defined in a relational DBMS.

It's unclear how flexible tomorrow's message stores will be. True, the ability to enter arbitrary documents is very attractive. But there are important disadvantages; eg, searching is harder and slower.

SQL Access

SQL, or *Structured Query Language,* is the data-definition and access language of choice in the DMBS world. The messaging world has only just started to support SQL, and Oracle Office and Notes are among the first products to do so. No doubt this will become much more common.

FIGURE 4 A FILE SYSTEM MESSAGE STORE
amo■



Older message stores don't use a DBMS. Eg, *NetWare Global MHS* stores messages in a hierarchical file system. Messages and attachments are stored, by user, in special directories.

Immo=