

Ceramic Technology International 1995

Are
your
designs
also
up-to-date?

RESEARCH AND DEVELOPMENT

136 Materials testing with monolithic and composite ceramics

Peter Gallon, *Instron Limited*

139 Analysis of ceramic materials using slurry sample presentation

Dion Tsourides, *Spectro Analytical Instruments Incorporated*

143 Particle size analysis — present and future

Chris Warran, *Roth Scientific Company Limited*

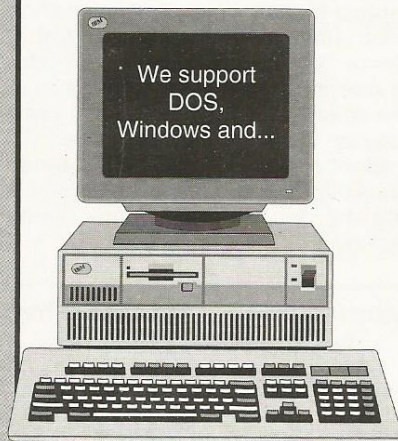
148 Powder and particle characterisation: quality control techniques

John Burmicz, *Quantachrome Corporation*

Software for Ceramic Science

The ceramic database you have needed for so long...
is minutes away by modem. Call our BBS now and download information or a free trial.

Ceramic industry, potters, universities and researchers in 30 countries have been using our software for 8 years. Now it's your turn.



FORESIGHT

from \$399

Physical Properties Database

Ideal for maintaining physical properties data. Manage all your recipes, tests, procedures, test definitions, materials, suppliers and contact data in one seamless relational system. FORESIGHT is fast and compact, it can communicate between sites, and it uses efficient FoxPro format databases for compound indexes and text storage. You can store a lifetime of information on one FORESIGHT equipped palmtop computer!

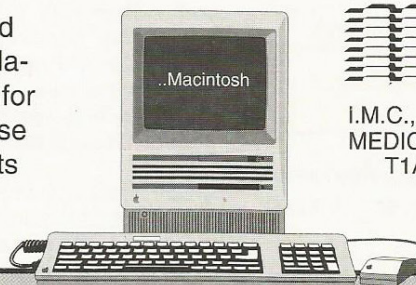
For serious record keeping and calculations, FAX or write for information on these and other products today.

INSIGHT

from \$159

Interactive Calculations

Ideal tool for formula and oxide level decision making. Instant formula to batch, batch to formula, formula to analysis, and oxide and material level property calculations. Materials stored in database, recipes in separate files. Supported by our newsletter in its 6th year.



I.M.C., 134 Upland Drive
MEDICINE HAT, Alberta
T1A 3N7 Canada

Phone 403-527-2826
FAX 403-527-7441
BBS 403-527-6074

Database technology to empower ceramic professionals

Profound developments that have empowered DOS & Apple Macintosh personal database users hold great promise for ceramic technicians. But there are basic issues concerning data format, hardware and software, and file structures to be addressed if we are to seize the exciting opportunity to form open and accessible worldwide ceramic information systems.

By Tony Hansen

Tony Hansen has been involved in the development of ceramic chemistry and database software at IMC and is manager of R&D at Plainsman Clays, a prepared clay body mining and manufacturing company.

Companies in many parts of the world have standardised desktop personal computers (PCs) much more powerful than what even software engineers were using just five years ago. Nineteen-eighties workstations which changed the industrial world are now considered toys compared to compact disc, modem and sound-equipped 486 and 68040 computers with $\frac{1}{2}$ gigabyte hard drives and 16MB of ram. Now the PowerPC, Pentium and palmtop models make even these appear backward. All of this is happening at prices no one could have imagined a short time ago.

While there are also revolutionary advances occurring in the workstation world, they are distant to the average technician contending with the gathering momentum of the PC and information industries.

Despite the abundance of literature chronicling the changes, most individuals riding this river of technology find themselves ill-equipped to navigate alone the issues that perplex even trained computer professionals. A torrent of software updates, new file formats, intimidating communication products, competing information technologies and advice confront us.

Nevertheless, there is no denying the potential of the latest generation of computers and software. Although there are many significant new PC-related technologies, the recent rise of low-cost mass-market relational database managers has the potential to touch the ceramic industry in a profound way. Their remarkable capability and ease of use give users a sense of power and control to accumulate and manage private worlds of recipes, materials, references, contacts, procedures, test data, and notes information. Most important, their wide availability holds promise for standards.

It is true that in the information industry traditional local or LAN multiuser relational database technologies are giving way to SQL, client-server, and object-oriented databases. Nevertheless,

these new trends are as yet having almost no impact on the consciousness of the typical PC user.

For the past decade, Plainsman Clays has developed software to put the power of ceramic chemistry within the reach of even nontechnical PC users. However, with the rise of powerful database development tools, we have also moved toward multilevel management of physical properties information related to recipes, materials and oxides. In addition, we are developing standards for file schemas, import and export formats and software for interdatabase record exchange.

Sharing database information. No one denies that standards are a key to co-operative progress. Without them in the ceramic database world, each user is an island of data surrounded by an ocean of obstacles to sharing it with others. However, one bright spot is that the difficulties in crossing this ocean are now much less hardware-related than in the past.

In fact, we have had a revolution in hardware technology that has brought diverse computing platforms ever closer. A standard Apple Macintosh can read and write DOS discettes, be networked with DOS machines, and now with the advent of the PowerPC, a stock Macintosh can run DOS software. The IBM PReP compliant PowerPC machines will appear soon and these execute a host of different operating systems, including Macintosh System 7, Windows and DOS. Thus dependencies on specific hardware or software vendors are becoming less important, especially if application software does not place excessive demands on system resources.

In addition, modems have made it possible for computers to easily establish a physical connection. However, there are considerable software challenges for meaningful talk between database systems. We have already built information highways for interchange of files and messages and we are used to exchanging spreadsheets

Am I pumping information into systems that will not easily give it up if, or when, they become obsolete?

and text documents. But databases are quite a different matter. They are structured according to the nature of the information they maintain, and the structure is inseparable from the data.

Furthermore, information is typically organised hierarchically in separate linked databases, each of which can be 10s or 100s of megabytes in size. Thus desired data must be extracted to an intermediate transaction file by software that understands the source database schema. This transaction file must then be posted to the target database schema by a program that understands the transaction file format and key mechanism. Failing that, the two sites must have the same database structures and schemas.

By standardising a scheme for ceramic databases and a universal intermediate tagged text format that respects parent-child relations, it will be possible to open a whole world of information exchange (Figure 1). For example, it would be nice, would it not to call your material supplier and download up-to-date technical information on materials they stock relevant to your use and then easily insert this in your private database system for permanent reference.

It is important to appreciate that the power and freedom to individually control and archive information comes with increased responsibility. Information is a valuable asset and it is potentially much more fragile on computer. Total

or partial loss is only a hard drive failure, computer burglary, database virus, or a forgotten encryption code away. However, I am most concerned about other questions: will my data format live as long as me? Am I pumping information into systems that will not easily give it up if, or when, they become obsolete? Am I employing effective database structures and relations?

Critical choices you make now will determine the lifetime of your data or its early demise in a morass of changing file formats, obsolete software or improperly designed file structures and links. Let us consider a few now.

A stable data format. New hardware and graphical operating systems have focused attention primarily on the user interface. This has tended to obscure the importance of the underlying data structures the software exists solely to create and manipulate. Applications live and die at the whim of market forces, but the key to the data's lifetime is its physical format. Widely used time-tested formats are vital if people in the mainstream wish to become more than just information packrats who cannot find or share what they have.

Database standards vary greatly in popularity, robustness and documentation. The quest for a proven, universal and reasonably robust format leads straight to the Dbase family. Dbase II was first introduced by Aston-Tate in the early 1980s and the Dbase physical format has grown with the times and is employed in many new Macintosh, DOS, Windows and UNIX products. It is thus interoperable, open, stable and documented.

DOS as a hardware platform. Since the Dbase file standard is independent of the operating system, a mixed Windows/DOS/Macintosh approach to

Field	Len	Type
description	50	char
date	8	date
notes	100	char
material 1	20	char
amount 1	8	num
unit 1	6	char
material 2	20	char
amount 2	8	num
unit 2	6	char
material 12	20	char
amount 12	8	num
unit 12	6	char

Table 1. A recipe structure.

maintaining a networked ceramic database is entirely feasible. Graphical applications, for example, have proven ideal for data entry front ends. While it is true that general purpose Microsoft Windows database software like Paradox and Access are very powerful, they can be sluggish in performance even on 486 machines. DOS, on the other hand, is fast even on machines far too humble for running Windows.

Our research and development has always been predicated on the assumption that speed of operation is extremely important, that there is no compelling reason to use slow or disc-hogging software. DOS is still a very important player in the database world; this market segment is the slowest to migrate to Windows. Most important, DOS is now a common denominator in that all major operating systems, including UNIX, can either emulate or run it directly.

Dbase software platform. Consider also that the maturity of DOS has spawned proven database products. Good examples are Computer Associates Clipper and Microsoft

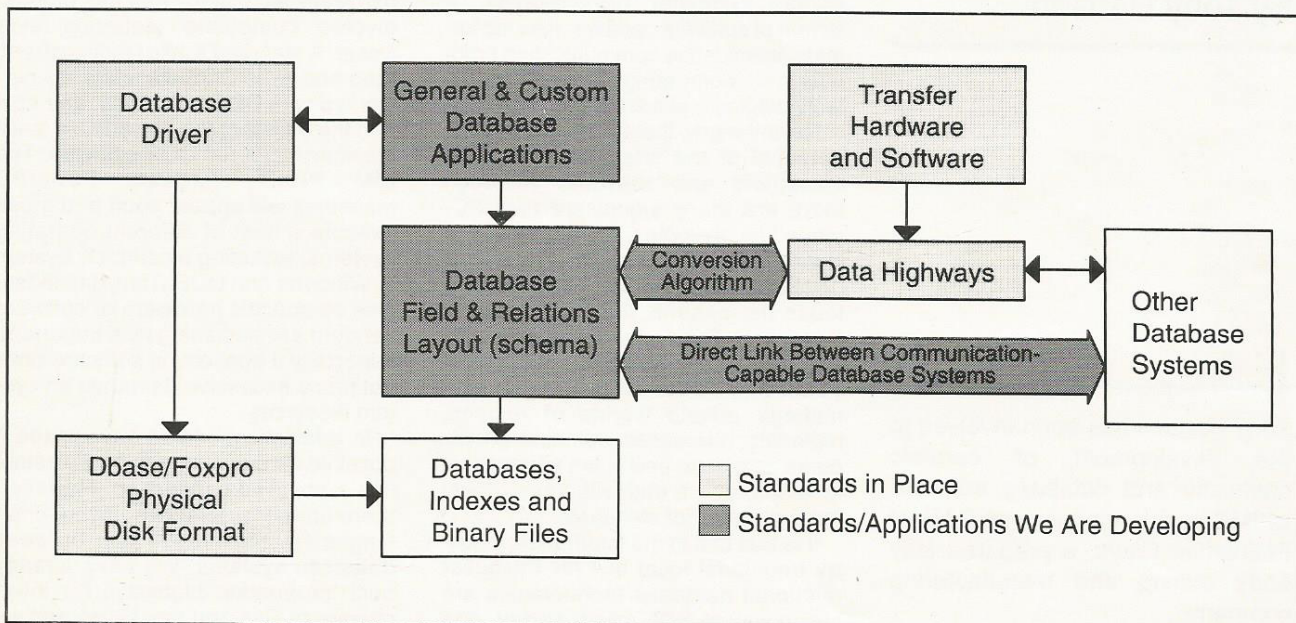


Figure 1. The state of ceramic database communication.

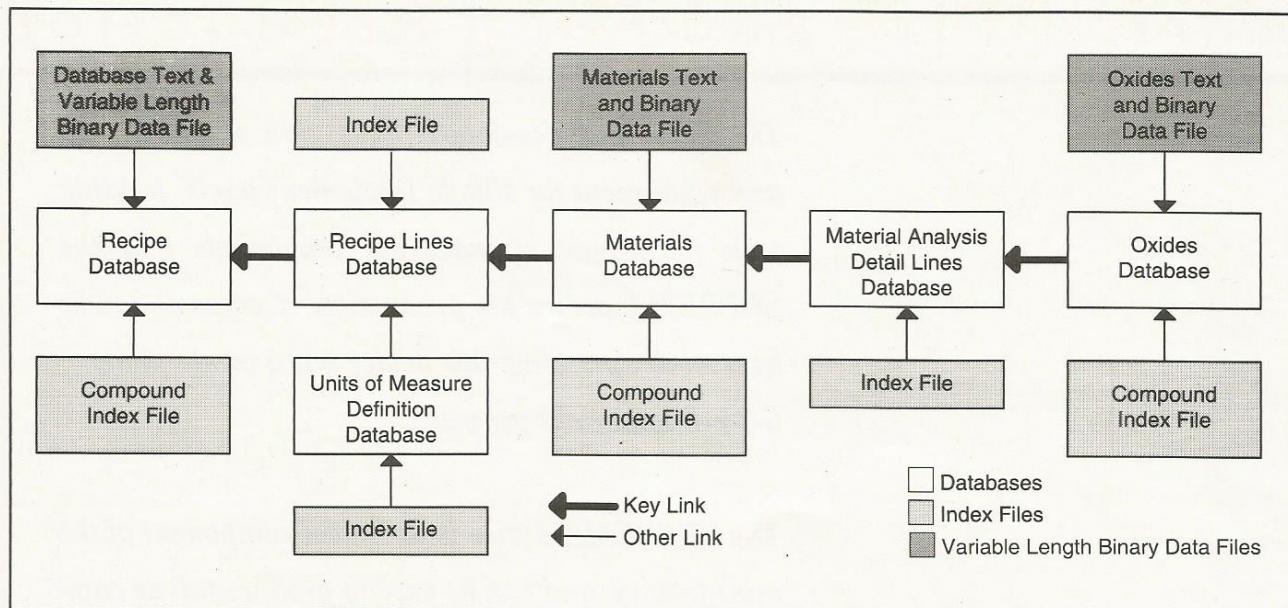


Figure 2. Fragment of relational schema for recipe storage.

Foxpro. Both have more than a decade of maturity and active followings worldwide through user groups and third-party support libraries, linkers and other products. Clipper produces compact quick standalone DOS programs for easy emulation on other platforms; Foxpro operates across DOS, Windows and Macintosh to interpret common source code. Our company has standardised on Clipper.

Logical data structures. Even the best database software and physical formats are ineffective if they are shackled with the wrong logical database structures and relationships. Typical ceramic-related information presents a number of unique database challenges that require thought and planning. For example, the recipe's real world presence is a study in hierarchy of information. Recipes have materials, materials have amounts, amounts have units, units have definitions, materials have variants, variants have oxides, oxides have elements, elements have definitions.

Add to this the fact that material variants have suppliers, recipes have tests, tests have definitions and procedures, tests have test data, and that recipes have properties. Faithful mirroring of the one-to-many and many-to-

one relationships are a natural application of separate but related database files.

For example, consider the task of storing a ceramic recipe in a database. A common error is to attempt to embody information in too few databases. For example, an inexperienced user might set up a recipe structure similar to that forced by a flat-file manager as shown in Table 1.

The approach used in Table 1 has a number of disadvantages:

- Since it is not hierarchical, it does not faithfully mirror informational relationships and does not easily fit into a larger schema to manage material, test data, procedures and so on
- The notes field is stored in the database rather than a separate variable length file. This is both wasteful for short notes and restrictive where more are required
- Complex procedural constructs are needed to maintain, change, report, and calculate field and record data
- It is restricted in the number of materials and wasteful of space for short recipes, and
- It makes for a difficult import destination without complex algorithms to distribute data among fields.

Even relational databases have tended to foster this type of approach among most nontechnical users because of the abstract nature of implementing relational mechanisms.

A fragment of a schema for representing a recipe and related oxides and materials is outlined in Figure 2. Text is stored in binary files handled by the database driver. Compound indexes allow multiple indexes for each database. The linkage between files is achieved using a unique three-character key code which is generated in a parent whenever a first child is added. The key uses upper and lower case alphanumeric digits to give a possible 62^3 possibilities.

This schema takes advantage of relational power. In addition, it employs a binary field which can hold text or any other appropriate data and multiple indexes which are maintained in a single compound index file.

Summary. While database technology is certainly not new, we now have an opportunity as never before to customise new mass-market database products to the ceramic industry, to store and share information in new and exciting ways. But that will only happen if we can establish standard database and transaction file formats. □