Getting the Most From PC Analysis **John Miglautsch**
DM News **Page 24**
December 1993

You know the data is there, but how do you get it out? Data is a byproduct of the direct marketing process. If you mail tens of millions of catalogs, you should have more than a few customers and orders. Gaining marketing advantage with the data can be more than a little challenging, however.

The first principle of data access/exploitation is that marketing must focus on the future. This means that data is helpful only if it can help in making decisions.

Principle No. 2 is that you should be sure about what you'll want from a marketing system. If you're not sure, you'll find yourself in some conflict with MIS professionals - you ask for twice as many things as you think you'll need and they deliver half the things you request (usually, the wrong things). Into this struggle steps the database marketing consultant, who listens to the marketing people as they explain how they want access, that they simply want to look into the data they know is there.

Attempts to settle these conflicts have included the use of access tools - slick point-and-click query instruments allowing unskilled users to pull complex selects and send output to Lotus, for example. But that wasn't enough for the marketing people (which may tell you something about the game we're in).

One query tool is designed to access a range of databases and only requires the user to learn one software package. To deliver this flexibility, power is often sacrificed. One particularly sophisticated tool worked well for my company until we started pulling down more than 100,000 records. The server selected the records and started sending the answer data in about an hour...then it took another 20 hours to write out the ASCII files from the PC. As the data expanded, the slick tool became unusable.

The initial reaction to the access tool is one of amazement and enthusiasm. Soon, though, the we'll-take-whatever-we-can-get attitude changes to it-has-to-do-this-and-that-or-its-really-no-good. As the demands are refined, it becomes harder and harder to execute specific procedures within the open-ended query environment. Surprisingly, there is a need for something like programs.

The situation is further complicated by the very nature of data. Few realize how much information is lurking behind the scenes when a routine report is generated. Product numbers are combined by product types, types are grouped, groups are sub-classed and subclasses are broken into divisions. Somewhere all that structure needs to be entered and maintained.

Data pulled directly from the mainframe usually lacks the additional organizational information. Users fill their PCs with enormous files of part numbers, but in frustration realize that they don't have all the pieces. Reports assumed to be part of the system are usually generated from offline PC tables - which means the information isn't even available to the rest of the organization.

The bottom line is that marketing pros aren't really interested in access (though that's precisely what they're

asking for). They don't really want to take the time to understand the nuances of the file structures or become adept at writing the countless macros needed to get data into a form for further analysis. What marketing really wants is the ability to ask a simple question and get a complex answer out of a number of analysis tools, ranging from statistical to graphic to mapping applications. They want answers, not access.

UNANTICIPATED QUESTIONS

Because marketing deals with the future, the primary requirement of a marketing analysis or decision-support system is the flexibility to ask (and answer) unanticipated questions. A system designed around a fixed number of variables and menu screens will be obsolete by the time it is created. In addition, we have also seen that simply retrieving data is not the final solution. Marketing needs data transformed and transported into reporting applications. This processing yields the answers needed for decision support.

The transformation system can be broken into three key components: the database, the questions and the applications. It is crucial to understand that both ends of the system, the database and the applications, are relatively constant. Only the questions change on a moment-to-moment basis.

Your database is founded on what's available from your main operations computer. This would start with a master file containing customer number, name, address, original sources, some sales history and sometimes demographic overlay data. Transactions are often split into order header and detail files. The header contains fields such as order number, date, keycode, territory, etc. The detail contains item-number, description, price, cost, units, shipping charges, etc.

Sometimes these two transaction files are combined. In addition, inventory master files can provide product-organization information - linking part number with product types, groups, classes and divisions. For catalog companies, it may also include ad-space information. Finally, there is usually a promotion file containing keycode information - promotion date, cost, description, etc. There may also be important organizational data allowing the grouping of keys into advertising type, campaign type and offer type. These five files provide the building blocks...and since data doesn't appear out of thin air, until new types of information (fields) are stored in the database, it remains constant in structure, only growing as more customers and orders are added.

The applications, on the other hand, require specialized types of data. Usually, almost any tool will accept ASCII files and may also directly access Paradox, dBase, .WK1 and .XLS files. Mapping tools require ZIP to ride along, while some statistics programs work better with continuous variables and others need categorical values. Most tools require extensive definition files that MUST match the data coming in. These definition files can be well over 100 pages of carefully structured information (and one stray character can blow the whole system away). It is precisely this complexity that makes it difficult to get answers out of the database.

Further, as understanding of the applications tools increases, more of the transformation work can be on the main database. Sales can be broken into sales by product category, and sales by product category can be grouped into categorical scores, which in turn can be selected directly and thrown into CHAID for analysis. It is far faster and easier to build these extra variables into the database than to calculate them every time they are used. Database design poses a dilemma: Until you build the database, you won't learn the tools - and until you learn the tools, you won't be able to design the database.

At this point you may want to consider getting help from someone who has done some of both parts. It will save you months of trial-and-error hardship.

As consultants, we have learned to build these systems by beginning with data samples. This allows us to handcraft quickly the various files into the structure. No two systems are the same, but there are common threads. For instance, it's important to realize there are only a handful of predictive variables within the database. Once these are identified and/or constructed, all further analysis will use the same ones over and over.

Each field is reviewed. Often there are wonderful field names in the files like "item category" or "customer type." When the values in these fields are listed out, perhaps 80 percent are "99" - miscellaneous.

Sometimes there will be wonderful distribution of values, but they won't match an equally wonderful set in another file. So the orders include a wide range of keycodes in the promotion-history file...but only 50 percent of the order keys match the promotion file (because the system does not check to make sure the order is entered with a valid keycode). Development tools like SPSS can greatly speed up the data-audit process.

Example analyses are crafted for each application. Sale buyers might be compared with full-price buyers, individual-item purchasers are compared with non-purchasers. These analyses' results might be thrown into mapping to see if the buyers live in different neighborhoods, or they might be put into CHAID for segmentation analysis. The point is that each application will require the information in its own format. Getting each application working with the data sets is at this point an exploratory process; you won't even have preliminary success until you've made several stabs.

Assuming the initial samples are close to acceptable, new specifications are generated for improving the applications power and building a phase-two database design. It is at this point, when the database is at about 90 percent of its final form, that it becomes possible to build the middleware. Middleware begins with development software. My company uses Paradox because it allows users to talk to the database (like a query tool) but allows developers to write programs that keep all the pieces straight.

The essential parts are scripts (which define available files and fields), the answer tables (generated from scripts), the reports (which process the answer tables into usable ASCII data files) and the definition files (which allow the application software to use the data files). In addition, the user interface is all point-and-click for simplicity. The ad hoc specification is met by allowing users to employ any criterion (including complex Boolean logic) for any relevant field. Users cannot make changes concerning which is returned from which fields, because that would scramble the pieces.

ANSWERS AREN'T ENOUGH

So, after building your analysis system, you now find all sorts of new variables that seem to help your business. You go back to your mainframe, which has the most current records and the big reel-to-reel tape drive, and you try to pull names for your upcoming mailing. Suddenly you realize that the key variables - the ones you'd thought made such precise predictions of who would buy - do not even exist on the mainframe.

An acceptable system cannot just perform dazzling analysis. It must allow you to take action and select on the transaction and other key variables.

Well over a dozen mailing packages allow you to predetermine counts for mailing, but in general they also require that you predetermine which variables you want counted - a far cry from the flexibility I described above. There are also "black box" systems that tell you who is best to mail, then pull the names for you. Though these provide analysis capabilities, they seldom give breakthrough insight into who's buying what or

how to expand your business. The full ad hoc analysis must be carefully connected to the list-selection capability.

Following the phase-two system development, we finalize the database design and respecify the application requirements. By now, we have a good idea of what should be included...and the most important predictive variables. This understanding allows implementation of the list-select applications. Since the same principle applies to this development - the principle that the marketing department doesn't know exactly what it will need down the road - care must be exercised to ensure maximum flexibility. However, unlike the analysis system, when pulling millions of names from a system, speed must also be an important consideration.

List selection is a very procedural process. It requires certain elements to be selectable (e.g.. RFM scores, ZIP and/or customer type, etc.). These should be done with buttons wherever possible. And there should be safeguards (decoy names, for instance) to prevent the selection of bad-debt names; a mechanism to check for accidental internal duplication; and a means by which users can split and chop selected segments to ensure testing and precise mailing quantities.

To accommodate the ad hoc needs of marketing, a query-style template can be included in the list-selection system. This query option cannot become the backbone of the list-select system, however. Being able to use any variable - even when buried within millions of transactions - will inevitably take some time. Draw a distinction between variables that will be used over and over on a production basis and those that will only be needed occasionally.

The list-selection system is the most procedural part of the marketing database. It must combine real programming power with recognition of the marketing people's specialized, unanticipated requests. For that reason, it is best designed with the marketing-analysis system.

Whether you build your database on your mainframe or your notebook, it must still satisfy these simple requirements: It must be flexible enough to allow you to ask any data-related question. It must be rigid enough that users can't easily break it. It should port data to a variety of applications. It must support regular list selection. Put 50-million records in it, put it all together and you have the makings of a world class database.