DATABASE RESEARCH GROUP, INC.

DATA BASE newsletter

Ronald G. Ross
Editor/Publisher

VOLUME 14, NUMBER 5
SEPTEMBER/OCTOBER 1986

IBM, On the Record Again!

An Interview with

Lois Dimpfel

Manager, Data Systems Products IBM, Santa Teresa

Lois Dimpfel

Recent performance improvements in DB2 have reopened continuing questions about the future of IMS and DL/I, and related products, as well as about IBM's positioning of the respective database products. Meanwhile, Fast Path is making an important bid to become a mainstay high-end database capability. Adding to the current confusion has been misleading commentary in the trade press concerning the possibility that DB2 and the much-anticipated repository might become required components of MVS.

In the following exclusive **Newsletter** *interview, Lois Dimpfel, who has management responsibility for both the IMS and DB2 product development groups, goes on the record for IBM once more to set matters straight. In doing so, she gives a sweeping view of requirements as IBM sees them for database product development over the coming years.*

Newsletter: It's been said that by the early 1990s IBM's database system products must support a transaction arrival rate of perhaps 4000 transactions per second. Is that range achievable within IMS?

Dimpfel: This is a requirement customers say they have, especially for systems that support consumer transaction processing. We recognize the need to support this requirement. We believe in particular that Fast Path could be used success-

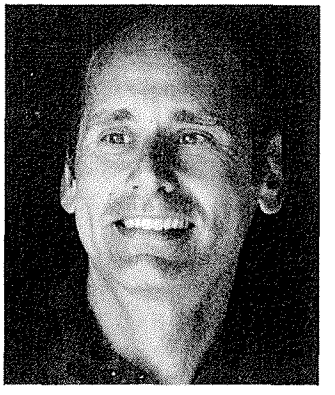
"Within IBM today, the investment that we are putting into the IMS product is roughly equivalent to the investment being put into the DB2 product. I think this clearly demonstrates that IBM anticipates *both* products to be around for many decades to come."

Continued on page 12

A Taxonomy for Productivity Software

By George Schussel

Digital Consulting Associates, Inc.



George Schussel

During the last ten years, the marketplace for productivity tools has changed significantly. Today, the market for such software products is large and complex. Instead of simply DBMS versus DBMS, the typical software vendor today sells a broadly functional, integrated set of software. Some buyers also want tools for the Information Center, whereas others wish to build large-scale production systems. Still others have eschewed the mainframe entirely and are building applications on stand-alone or networked micros.

This article looks at the current market for high productivity software and comments on the different categories of software available in that marketplace.

1. Integrated Development Software

This is "major league" software—the guts of the 1980s EDP department. It usually consists of a DBMS, an integrated

". . . the computer performance of a system developed with a 4GL may suffer compared with a comparable application developed in a COBOL/CICS environment, but this performance degradation is normally no worse than 100%."

active data dictionary, query language, transaction processing monitor (often CICS), report writer, micro-to-mainframe link, and interfaces to various other packages frequently including applications. A key portion of the integrated software package is the data dictionary/directory, which controls data definitions and is therefore primarily important in coupling the different software pieces.

Examples of products that compete in this market are Cincom's TIS/XA-SUPRA, Cullinet's integrated software line based on IDMS/R, Applied Data Research's DATACOM/IDEAL, Software AG's ADABAS/NATURAL and Computer

Continued on page 5

addition of a "null value" to the domain of the attribute or by the use of another attribute to indicate the presence of null values in the attribute instance.

For example, in the MARITAL-STATUS situation, the choice of "blank" or some other code to indicate "marital status unknown" is probably harmless and is, in fact, usually the method employed. This technique fails for those attributes (such as numeric fields) in which all possible bit combinations are members of the domain of the attribute, preventing the addition of a "null value" indicator. In this case some other, existing attribute must be used to indicate a null condition, or an attribute must be created to do so.

Physical Compromises for Structural Nulls

Entity state nulls may be handled in a manner similar to non-structural nulls. Either a value can be added to the domain of the attribute, this value indicating "null," or another attribute may be used to indicate the existence of null values in the attribute.

Entity type nulls are handled in basically one way, that being the creation of a new attribute, the "indicator attribute," to show which of the following sets of attributes is not null. This is the most "foolproof" method; there is no possibility a single "indicator attribute" could contain conflicting values. If the existence of "non-null" values in the individual attributes were used to indicate which set was valid, the probability that ambiguous information could be conveyed would be high. For example, in the ANIMAL example above, a new attribute, TYPE-OF-ANIMAL, might be added to the entity ANIMAL. The domain of this attribute would be [F, H], where F would indicate "finned" and H would indicate "hooved." If an "indicator attribute" was not created, it would be possible that both TYPE-OF-HOOF and TYPE-OF-FIN could both be non-null, making it impossible to determine which was the correct value.

Conclusion

The existence of nulls in a logical model is not only confusing and ambiguous but also requires that the number of syntactical constructs necessary for complete representation of the model be increased. This raises the probability of misinterpretation of the logical model.

The existence of nulls in a physical model is not only allowed but is sometimes required for performance reasons. However, whenever such a compromise is made during the creation of the physical from the logical model, it must not only be thoroughly justified, but also clearly documented. ■

SCHUSSEL from page 1

Corporation of America's Model 204/User Language. All of these products run in the IBM 370 architecture, and some also run under the DEC VAX/VMS operating environment.

Additionally, other mainframe companies offer comparable software product lines. Examples include Burroughs with DMS II/Link II, NCR's TRANPRO/TOTAL/MANTIS, Sperry's innovatively different MAPPER, and ICL's IDMS/QUICKBUILD.

Integrated development software is normally effective only on a machine that is at least as large as a large mini-computer (i.e., IBM 4341 or larger). Such a software set is certainly recommended if a company is developing transaction processing systems as applications. Alternatively, if the company has large databases and wishes multi-user applications to be supported in a mainframe environment, this is also the type of software required. Most companies that are developing corporate-wide applications, or what they consider "strategic systems," will use integrated development software for that development.

Building applications with these product sets is not cheap. The software alone typically runs $300,000+ for a paid-up purchase license from independent software vendors. Regardless of the hardware/operating system architecture, the installation and use of an Integrated Development Software set of tools require a technical staff that includes, but is not limited to, Database Administration. The training time required to become educated in the use of the products, and to develop Database Administration and support, can easily require a year or more.

One important compensating advantage, however, is that all of these software tools contain powerful fourth generation languages (4GLs) that are useful for replacing the majority of COBOL code with much higher-level programming.

It is interesting to observe how IBM, the largest software vendor of all, participates in this market. IBM offers various software components that have competed in this marketplace. Examples include its database management system (IMS), data dictionary (IMS DB/DC Data Dictionary), fourth generation language (ADF II), and so forth. In general, these IBM software products are often viewed as being technically in-

About George Schussel

The author is founder and President of Digital Consulting Associates, Inc. (DCAI), a consultancy based in Andover, MA specializing in software productivity tools, including DBMS, 4GL, and Artificial Intelligence. DCAI creates and sponsors seminars and conferences about the software industry. Through his participation in these conferences, the author is continually studying, analyzing, and classifying various types of software products. In this "Taxonomy" article, he explains the classification scheme presented at DCAI's National Database and 4th Generation Language Symposium.

Dr. Schussel received his Bachelor's degree from U.C.L.A. and his Master's and Doctoral degrees from Harvard University.

Upcoming DCAI conferences with a software focus include the following:

National Database and 4th Generation Language Symposium

Dallas	October 27-30, 1986
Boston	December 2-5, 1986
Melbourne, Australia	February 17-20, 1987
Wellington, NZ	February 23-26, 1987

Software Trends for Executive Planning and Strategy

Las Vegas	January 27-30, 1987

National Data Communications Conference on Local Area Networks and Micro-to-Mainframe Links

Washington, D.C.	December 9-12, 1986

ferior to the programmer productivity tools available from independents. The principal reasons usually cited include their lack of user friendliness, the lack of integration among the various products, and the lack of active or integrated data dictionaries.

The recent emergence of new IBM products, including SQL/DS, DB2 and CSP, indicates that IBM wants to be competitive in this marketplace. In terms of programmer productivity, these new software tools, especially DB2, are an important improvement over earlier products. But they still suffer in the area of integration and thus far lack integrated, active data dictionaries. They also do not generally present common user interfaces and approaches.

2. Mainframe-Oriented Programmer Fourth Generation Languages

When the 4GL component of an integrated development software set is coupled with a standard file management system (rather than a DBMS), the result is a programmer-oriented 4GL. In other words, when the DBMS, integrated data dictionary, report writer, query language, etc. are all eliminated, the result is a screen-painting productivity tool. On a stand-alone basis, 4GLs still therefore provide powerful productivity enhancements in the development of screen-oriented, transaction-processing applications.

Examples of these products include Cincom's MANTIS, Software AG's NATURAL, McCormack & Dodge's MILLENNIUM, and MSA's INFORMATION EXPERT.

While all 4GL programming languages offer important productivity advantages over straight COBOL/CICS, there are many different "flavors" of programmer-oriented 4GLs. Some are quite procedural and language-like; others are functional development tools using menus, dialogues, or other newer types of user interfaces besides programming.

"A number of COBOL System Generator products can generate code with better execution time than all but the top 5% of the typical programming staff."

Programmer-oriented 4GLs are designed to implement systems even with complex business logic and large-scale TP volumes. Whereas the computer performance of a system developed with a 4GL may suffer compared with a comparable application developed in a COBOL/CICS environment, this performance degradation is normally no worse than 100%. (This may not sound very good, but remember that the productivity enhancement for the programmers actually writing the code is probably 500% or more.)

Programmer-oriented 4GLs should not be confused with higher-level non-procedural languages such as FOCUS and NOMAD2. These latter products are more directed to Information Center and end-user use, and typically generate systems that perform three to eight times more slowly than a comparable COBOL system.

Programmer-oriented 4GLs are worth investigating if the company:

- needs to develop screen-oriented transaction processing applications, *and*
- has trained programmers, *and*
- wants more productivity, *and*
- prefers to spend $60,000-$100,000 for a 4GL license fee rather than the $300,000+ for a full layer of integrated software.

3. COBOL System Generators

COBOL System Generators (CSGs) provide a different type of programmer-oriented 4GL environment. They differ from the products in the former category because CSGs generate COBOL code and source calls for database or file structures rather than final object code. Examples of products in this category include Transform Logic's TRANSFORM, Pansophic's TELON, and CGI's PACBASE.

COBOL System Generators provide an interesting contrast with programmer-oriented 4GLs. They both compete for the same application development dollar but with different relevant strengths.

The most full-function COBOL System Generators offer a quite broad range of services including data dictionary, automatic documentation, project control, screen painting, automatic database navigation, DBMS call generation and test facilities. COBOL System Generators need to be complemented with standard software tools such as VSAM, CICS and IMS DB/DC.

If the company uses a CSG, a compile time from three to eight times longer can be expected compared with hand code. Execution time, however, may be better. A number of CSG products can generate code with better execution time than all but the top 5% of the typical programming staff.

The use of CSGs will provide an improvement in a shop's programmer productivity. Tests have shown that this improvement is in the range of two to one over straight COBOL. This is somewhat less than might be expected in a programmer-oriented 4GL environment, where four to one improvements are commonly reported.

4. Information Center Software

The tools in categories one through three are principally designed for MIS shops and professional programmers. They enable existing programming staff to become more productive.

Information Center software, in contrast, is software designed to convert non-data processing professionals into systems builders. Popular software in this category includes Information Builders FOCUS, D&B Computing's NOMAD2, Martin Marietta's RAMIS II, Infodata's INQUIRE, and Battelle's BASIS.

The "programming languages" provided in this category tend to be much more non-procedural. Frequently, menu-driven user interfaces are provided. Information Center software also makes extensive use of default options so that little specification is required if querying and reporting is the goal of the end user.

Information Center software is *relatively* easy to use compared to integrated development software. "Easy to use," however must be defined in relative terms: getting the Information Center ready to operate after initial installation of the product can still take as long as a year. The time can be well used, however, to create an information support organization, develop a small number of prototype applications, design the necessary training, and establish relevant databases for querying and reporting purposes.

Information Center software is used not only by end users but also by MIS departments. In general, the slow performance of this software (compared with COBOL) means that

Continued on page 10

SCHUSSEL from page 6

it is most suitable for departmental, not corporate-wide or strategic systems.

Information Center software has a reputation of high CPU and memory usage. Experience shows that the combination of these high computer utilization rates and the products' popularity for end user applications frequently result in a rapid escalation of computer resources devoted to this environment.

Information Center software works best with relatively static databases. These products are not designed for production transaction processing. They are typically single-threaded and query- and report-oriented.

Passive data dictionaries seem to be characteristic of Information Center software. Beyond the dictionary, a broad array of interfaces to many different foreign software packages, such as DBMSs, statistical packages, and graphics, is usually supported. Information Center packages such as NOMAD2 and FOCUS have been the first products from independent software vendors to support both read and write access to IBM's relational DBMSs, DB2 and SQL/DS.

5. Decision Support Systems

Decision Support Systems (DSSs) are software systems that include a combination of data modeling, DBMS, query facility, and simulation language. These combinations are well suited for decision analysis and correlation analysis at the field level. DSS products are normally used for modeling future scenarios rather than reporting against already created historical databases.

Simple two-dimensional spreadsheets like VisiCalc or LOTUS 1-2-3 can be thought of as Decision Support Systems. However, mainframe-oriented DSSs can be significantly more powerful. Comshare's System W provides an interesting example of a multi-dimensional analytical tool that provides more sophistication than a spreadsheet. Information Resource's EXPRESS combines multi-dimensional analytical facilities with a true DBMS environment. Many DSS vendors are now providing micro-computer implementations of their mainframe software.

DSS tools are frequently sold to end users in production, marketing, and finance departments. Their user interfaces are significantly friendlier than programmer-oriented 4GLs and typically do not require any programming.

Whereas DSS tools seem functionally similar to Information Center software, important differences do exist. On the one hand, DSS tools can be more easily used to model difficult, complex problems than Information Center tools. On the other hand, Information Center tools can be used to build production (transaction-oriented) systems, whereas the DSS tools are designed for databases that are essentially static.

If the company requires more power than LOTUS 1-2-3 offers, one of these Decision Support tools may be a good solution.

6. Design Automation Tools

As readers of the *Newsletter* certainly know, software productivity tools by themselves do not solve the problems of automation. The successful MIS department needs a methodology and engineering approach within an overall architecture in which these 4GL tools can be used. Software-based design automation tools can be considered the next generation of the structured methodologies that evolved in the 1970s.

In general, two different types of methodology software products appear to be evolving. One is the information engineering/action diagramming approach espoused by James Martin. A good example of this approach is the Knowledgeware/Arthur Young Information Engineering Workbench. The second includes normalization support software tools (such as Technology Information Product's FACETS and Knowledgeware's DATA DESIGNER).

7. Supermini Vendor Software

This category comprises productivity software sold by the hardware manufacturers; for example, DEC's VAX Information Architecture, Concurrent's RELIANCE PLUS, and Hewlett Packard's ALLBASE.

The penetration of software produced by hardware vendors (including IBM as a mini vendor selling the s/36 and the s/38) into mini-computer markets is significantly higher than the corresponding penetration by IBM into the 370 architecture market. Reasons for this include the relatively low costs associated with acquiring the hardware vendor's software tools and the fact that independent software companies with competitive products have only recently appeared in the supermini arena.

Because the various minicomputer architectures are similar, it is common to find independent vendor software products that run on the hardware and operating systems of several different minicomputer vendors. Customers who choose a hardware vendor's software therefore sacrifice portability to other vendor environments.

8. Supermini Software—DBMS and 4GL

Minicomputer operating systems offer friendlier environments, and greater ease of use, than the batch-based, mainframe-oriented MVS and VSE operating systems. As a result, many supermini systems are showing up as departmental machines or corporate data processing systems in mid-sized companies. Application systems built in these environments will play an important role in providing connectivity between widely installed PCs and mainframe-oriented databases.

This marketplace is served by independent software vendors who market integrated development software that usually includes a database management system and fourth generation language. Examples are Software AG's ADABAS/NATURAL, Cincom's ULTRA/MANTIS, Relational Technology's INGRES, Seed Software's SEED, Software House's 1032, and Oracle's ORACLE.

In contrast to comparable software products marketed for IBM mainframes, many of these DBMS/4GLs are based on more modern technology, but are less mature (i.e., offer less total function). The most common database model used in mini implementations is relational. This differs sharply from the 370 world where most currently installed DBMSs were developed before the relational model was widely adopted.

Several important products in this market (including ADABAS/NATURAL and ULTRA/MANTIS) have been ported from the 370 architecture. For these products, it is possible to use identical source code in both mini and mainframe environments. The vendors of these products also pro-

vide communications software to connect the minis and mainframes.

9. Supermini Software—Fourth Generation Languages

As in the 370 world, it is possible with minicomputers to buy a stand-alone fourth generation language for building applications that interfaces with a file system. POWERHOUSE from Cognos, APPLICATION FACTORY from Cortex, ALLY from Foundation Computer Systems (DEC's RALLY), and INFO from Henco are among the leaders in this category. FOCUS, which is now available in the DEC, Wang, and Unix environments, competes in this category as well.

Programmer-oriented 4GLs on minicomputers are frequently more modern and general-purpose than their counterparts in the mainframe world. Examples of some of the best mini 4GLs include ALLY, POWERHOUSE, APPLICATION FACTORY and PRO-IV.

10. Database Machines

Database machines are typically implemented as back-ends, the converse of front-end dedicated hardware/software computers. Database machines have the goal of offloading the database and disc file management from the mainframe. Implementation requires software that runs in the mainframe to interpret data manipulation language (DML) commands. These commands are then shipped down a channel to the dedicated database machine that manages the disc file environment. The two dominant vendors in the database machine marketplace are Teradata and Britton Lee.

Database machines may offer unique advantages in situations where there are dissimilar hosts, multiple computers, and/or very large databases.

In today's software implementations of the relational model, it is clear that relational DBMSs provide significantly less transaction processing capability than other types of DBMS implementations. When both a high transaction processing rate and a relational environment are desired, database machines may offer the best cost-effectiveness. Teradata, for example, has recently announced transaction processing speeds for its DBC 1012 that match IBM's IMS Fast Path.

11. Microcomputer Software—Single-User Systems

The best known micro software market is probably the single-user DBMS/programming language combination, typified and dominated by dBASE III. Other important competitors are POWERBASE, and R-BASE. A microcomputer DBMS is differentiated from a simple file manager by its ability to search for, and report on field values without reading complete records or searching complete files. A true microcomputer DBMS should be able to JOIN several files and should also offer a programming language. Although a range of power and user friendliness is represented across this product category, none of these products is really suited for the casual user.

12. Microcomputer Software—Multi-User Systems

The second microcomputer category includes multi-user microcomputer systems. These products may be explained as minicomputer software systems which were implemented for multi-user micros once these micros had adequate memory and operating systems to support them. Good examples are UNIFY and INFORMIX. These products add security, COMMIT updating, Database Administration utilities and concurrency controls to the capabilities of the single-user systems. They typically deliver a true DBMS (usually relational), a simple 4GL, report writer, query language, and sometimes a data dictionary.

13. Microcomputer Software—Integrated-User Systems

The third category of microcomputer productivity tool is integrated software/mainframe extensions. These products are multi-functional LOTUS/SYMPHONY/FRAMEWORK competitors. Many of them, like Cullinet's GOLDENGATE, ADR's PC DATACOM, and Computer Associate's EXECUTIVE are sold by mainframe vendors, not as direct competitors to LOTUS and SYMPHONY, but as a way of providing integrated services for microcomputers in shops that use that vendor's mainframe software. These products normally include a relational DBMS, a query language, spreadsheet, word processing and a micro-to-mainframe link to the vendor's DBMS. Better implementations allow the micro user to process mainframe DBMS data with transparency such that it appears to the user as though the data were located in the micro's own spreadsheet or data files.

14. Microcomputer Software—Micro-to-Mainframe Links

The final category of microcomputer software comprises the micro-to-mainframe link. Links are used to integrate micros with mainframes or minis, and to transfer files between them, so they can be processed wherever is more convenient.

Link technology available in today's market varies widely ranging from simple 3270 emulators to integrated, virtual DBMS interfaces that support both uploading and downloading. Link and workstation technology is relatively new and still lacks standards. It presents the typical user with a number of problems, including slow transfer speeds and low levels of security and integrity. After adding up all the costs of cabling, micro-and-mainframe software, additional hardware, and mainframe cycles, many micro-to-mainframe link buyers have found that the incremental costs of integration far exceed the costs of installing separate stand-alone hard-disk PCs.

In spite of the current high cost of linking personal computers, links, LANS, and other connectivity solutions will surely become extremely popular. In the US market it is predicted that by the end of 1986 there will be 800 business micros in use for each mainframe. As these systems require integration, they will generate a growing demand for reasonable products. Link-oriented products will therefore become a sizeable business. ■