

# Secret to cutting backlog? Write less code

By Naomi Lee Bloom  
Special to CWI

One almost universal, but only moderately successful approach to reducing application backlogs has been to improve the productivity of scarce technical resources. A more promising approach to meeting user application needs may be to reduce substantially the amount of new code needed to satisfy these needs. Such application-enabling techniques, to

Bloom is a senior principal of American Management Systems, Inc. in Arlington, Va.

*Other things being equal, the less code written to achieve a specific level of systems support, the less risk, cost, elapsed time and staff frustration. It is important to note, however, that other things are rarely equal.*

use a phrase that seems to have originated within IBM, are intended to reduce the amount of new code written, rather than merely expediting the production of new code.

Other things being equal, the less code written to achieve a specific level of systems support, the less risk,

cost, elapsed time and staff frustration. It is important to note, however, that other things are rarely equal. Many of the techniques described below substitute increased consumption of computing resources for reductions in the personnel resources needed to achieve a certain level of

user support.

There is nothing very mysterious about finding ways to write less code. You can:

- Convince the user not to want (or need) a new application.

- Reuse old code — your own or someone else's.

- Use simple tools (remember how levers work?) to multiply the work value of any code you write.

- Get someone else, perhaps your users, to write the code for you.

The key to successful application-enabling is to build these concepts into your systems development life cycle methodology. Custom development or even package installation projects should not be initiated unless the new application is really needed. And at every stage of the life cycle, you must ask yourself what alternatives exist to developing new code. Thus, application-enabling techniques parallel, in some sense, the systems development life cycle.

The most obvious solution to the problem of how to write less code is to eliminate from the backlog all but the essential applications. Strategic systems planning is the process by which an organization identifies and prioritizes its major systems development objectives. By explicitly aligning systems development priorities with the organization's business strategy, we take a critical first step toward reducing the amount of new code to be written. To repeat, if you don't develop unnecessary applications, you will not be called upon to write (and maintain) worthless code.

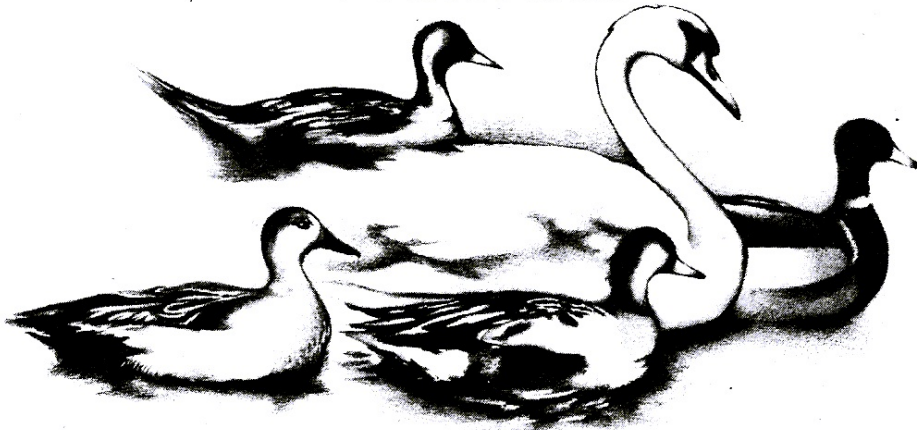
Where an application is justified, there are several possibilities for developing it without writing any code or by writing only a small amount of simple code. Application software packages have been available for nearly 30 years, and many routine business and system functions are very adequately supported by such packages. Before deciding that an application is so unique as to obviate using any existing code — a common attitude among many in-house analysts and users — consider the many flavors of software packages and reusable in-house code.

Currently available commercial applications and systems software can be divided into three general categories. Traditional software packages perform a well-defined set of functions with minimal installation options. Contemporary software packages perform a well-defined set of functions subject to many table-driven, user-defined and/or installation-specific options. And adaptable software packages perform a flexible set of functions subject to many table-driven, user-defined and/or installation-specific options.

Written to be customized, contemporary packages have well-documented source code constructed to support low-risk user modifications. Such packages also use reference tables to remove from the source code the most frequently customized functions, such as report headings and, in some cases, formats, message code literals and severity levels; data element names; field lengths; data types; edit rules (including pointers to other reference tables of valid values and code translations); and pa-

See CODE SR/50

## Why is it important to you that CONSCO's financial software is different?



Because CONSCO's financial software systems were designed for accountants by accountants. Today's end users need more information about their business operations than ever before. That's why we designed our systems using what we call "Financial Technology"—the ability for a user to enter, analyze, and report meaningful business information without data processing limitations or restrictions. The outcome—two unchallenged financial systems designed just for you.

### Accounting Information System

**A complete General Ledger system that can provide many more applications than just general ledger**

AIS was designed at the outset as a full blown on-line information system so your entire company can now be reflected in one integrated system. Now managers at all levels can get the information they need to really do their job.

You can even do such things as work order management, construction work in progress, cost control & analysis, project reporting, encumbrance accounting, as well as all those special things that just can't be labeled.

### Consolidation System

**Finally a financial system designed specifically for consolidation accountants**

CONSOL fully automates those time consuming and tedious tasks like reconciling intercompany accounts and producing management reports. This saves you time when you need it most—at the end of the month when you're under the gun. Spend your time doing meaningful analysis instead of number crunching. Not only will you get better results, you'll save money.

CONSOL's unique architecture produces results for you up to 200 times faster than our competitors. This means you get your information now.

**Why is it important to you that our company is different? Because you are different. And CONSCO cares about you!**

We send our people to you. We work with your data and your deadlines at your site to give you results that are put directly into production. We call this the CONSCO Insurance Policy because it insures that your job gets done. You'll see that our service is different too!

**If you care about you, the way CONSCO cares about you, call us at (212) 586-8701 or write**



conSCO

888 Seventh Avenue, New York, NY 10106





## CODE from SR/46

parameter values such as process scheduling dates, current withholding tax percentages and airline overbooking percentages.

The purchase and use of contemporary or adaptive application packages generally reduce the costs, risks, elapsed time and personal frustrations of meeting systems support needs. However, there is always a price for flexibility. Sophisticated reference tables can require considerable loading and maintenance effort. But this approach is far less risky than modifying source code, and users can often be encouraged to take responsibility for loading and maintaining most of the tables.

From the perspective of cost, risk and the containment of elapsed time,

the availability of options means someone must analyze, document, recommend, evaluate and decide upon each desired option. Contemporary application packages go a long way toward meeting organization-specific requirements without developing new code.

There are two general approaches to multiplying the value of any code you do write. Extension software uses your simple code written in the tool's own command language as the input from which it creates (by translation, compilation, assembly or one of several other extension techniques) very substantial functionality. Report writers are a common extension technique.

Conservation techniques are a formal set of design techniques that look for the common functional elements

in an application in order to develop a single implementation of these common functions for use across the application. Reusing date routines is a very simple case of conservation.

The universe of extension software ranges from the old and familiar to the new and still developing, including:

- Utility programs that provide system or housekeeping functions.
- Report writers and inquiry languages, including graphics packages.
- Data base management systems with which you use simple commands in the application programs to invoke powerful data handling, editing, storage and access capabilities.
- Screen generators.
- Data management and analysis tools.
- Application generators.
- Very high-level languages.

However, there is a significant problem currently associated with the use of extension techniques. Until considerable standardization occurs, taking advantage of even a small set of these tools will impose a serious training burden on your organization. Many professional programmers and users will resist using these tools because they quite reasonably perceive that the cost of mastering them is too high.

Perceptive analysts and designers have always recognized common functions in their application specifications, but the process of doing so has been largely informal. Many business applications contain a rather large set of common functions that lend themselves to a common software approach.

The decision to build an application around a base of common software modules must be made explicit quite early in the design process so that all further effort can be efficiently directed.

End-user computing is not a new idea, so why do we now treat it as a state-of-the-art development? One reason is that, until now, whoever approached the computer was forced to learn computerese — at great personal sacrifice. If we believe the advertisements for various end-user computing tools, the professional programmer may soon focus solely on core production systems and tool development, leaving to the user the development of most data extraction and analysis systems. Clearly, if the user can directly translate his unspoken information requirements into a working system, he won't have the DP staff to kick around any more.

If only to sell more computers, hardware vendors would welcome any approach to program development that used more computer resources to leverage scarce personnel to develop new applications that used more computing resources. Since they develop many of the packages and tools and generally corner the market on really superb professional programmers, software vendors certainly favor the techniques described here. Corporate users and DP managers are also on board the write-less-code bandwagon.

Time is clearly on the side of the approaches described here, but I would not yet discharge my Cobol programmers nor declare that all user needs can be satisfied by a new data base management system. As in all things, a balanced mix of these techniques with more traditional application development strategies will produce the best results.

## MANAGING INFORMATION FOR ALL ITS WORTH.

ISA has 13 information systems working profitably for nearly 2,000 companies worldwide.

These systems provide everything from solid control of accounting data to sophisticated financial modeling and portfolio management.

Our consulting staff leverages talent customers have in-house and assists with requirements studies, system development, even turnkey implementation.

We sell solutions. Our people talk business, not software.

ISA customers vote their confidence every year with the highest percentage of product support renewals in the industry.

So what are you waiting for? Call or write the ISA Information Center, 500 Northridge Road, Atlanta, Georgia 30338. In Georgia, call (404) 587-6800.

## CALL ISA TODAY.

1-800-554-1193

© 1984 Information Systems of America, Inc. A United Telecom company.

# ISA™

INFORMATION SYSTEMS  
OF AMERICA, INC.