EDP ANALYZER
Editor and Publisher: RICHARD G. CANNING

EDP ANALYZER

MARCH, 1982
VOL. 20, NO. 3

INTERESTING DECISION SUPPORT SYSTEMS

One thing that study after study has shown is—most managers do not themselves use computers for aiding in their decision making. Yes, they do receive reports from the regular data files that give summaries of activities and such. But managers have made little use of the computer for (say) exploring alternative solutions to problems. That picture is changing, however. New, powerful, 'friendly' decision support software has become available, as have the new personal computers. You will see a big growth in the use of these by managers in the next few years.

James B. Edwards is a professor of accounting at the University of South Carolina. He has a Ph.D. in business administration, he is a certified public accountant in three states, a certified internal auditor, a certified management accountant, and he has a certificate in data processing. Due to this background, he is often called upon to be an expert witness in tax and other litigation cases. In preparation for giving testimony, he uses a micro-computer and two decision support system packages to study alternative forecasts relating to the economic issues being disputed.

In 1977, when Radio Shack announced its first personal computer, the TRS-80 Model I, Edwards bought one with 4K bytes of memory. Since then he has enhanced the system so that it now has 48K bytes of memory, four 5-1/4 inch floppy disk drives (storing 358K bytes of data), and a 132-column printer. He also has Radio Shack's word processing package, Scripsit, a general ledger package, and two forecasting packages from Applied Economic Analysis Corporation, of Long Beach, California (Reference 10).

One of the packages is the Business Planning and Forecasting Package, a general purpose planning tool for use with time series data. It allows users to forecast sales, costs, interest rates, and so on. It contains a short term forecasting model, which uses exponential smoothing or adaptive filtering techniques, and has multiple regression and seasonal adjustment options.

ISSN 0012-7523. Photocopying this report for personal use is permitted, providing payment of $2.50 fee *per copy of report* is made to Copyright Clearance Center, Inc., 21 Congress Street, Salem, MA 01970; please include identifying fee code 0012-7523/82/030001-14$2.50

The package also has a data management facility for creating and modifying data files. New values of data calculated by the package (called transformations) are stored in new files by this facility, automatically. The package also has a graphing facility, files of thirty of the most widely used economic time series data presented quarterly from 1955, and a forecasting example which takes about fifteen minutes to run. This example was most helpful in becoming familiar with the package, Edwards told us.

The second package is called the Box-Jenkins Forecasting Package. The Box-Jenkins method of forecasting is a sophisticated technique for unraveling complex and inter-mixed patterns within time series data. The package tries eight combinations of patterns—different types of trends, seasonal patterns, and cyclical patterns—and then determines which combination best fits the data, and how closely it fits. Each of the packages costs $99 for TRS-80 Model I (or III) and $199 for Model II.

In preparation for his expert testimony, Edwards attempts to forecast interest rates, rates of inflation, sales, etc. For example, in several cases, automobile and truck manufacturers withdrew their franchises from dealerships. Edwards was asked to forecast the projected losses that these dealers would suffer under different economic conditions.

In another case, a group of homeowners brought a class action suit against the housing project developers because of a contract escalation clause linked to the consumer price index, which rose sharply after the contracts were signed. In this case, he evaluated several alternative settlement proposals by forecasting the consumer price index for a number of years into the future and then calculating the possible present values of those settlements.

For these and other types of forecasting problems, the two forecasting packages now perform 90% of his calculations, Edwards told us. Previously, he used a time-sharing service to perform such calculations. But since the cost of each forecast was quite high, he limited the number of alternatives to as few as possible. With the micro-computer, he does just the opposite—he tests all of the alternatives he can think of. Since he received the packages in December 1980, he has used them over 500 hours.

Another benefit is that the Box-Jenkins package allows him to uncover complex trends which were virtually impossible to find manually. Generally, he uses five years of monthly data. He specifies whether the trend is upward, downward, or horizontal; and then he lets the package look for the best fit. For this amount of data—60 data points—the program takes about ten minutes. He lets it run while he turns to other work.

Edwards is very pleased with these two decision support packages. He is especially pleased that they were supplied in source code, because he sometimes makes a few temporary changes to tailor the packages to forecasting problems. For example, he has extended one package to permit forecasting 120 periods into the future. These packages allow him to perform numerous complex computations at a very low cost. And he feels the results allow him to offer more informed opinions.

International Harvester

International Harvester is a large manufacturer of heavy trucks, agricultural and construction equipment, and components (such as engines and transmissions). They also have a large financial services division. IH had sales of $6.3 billion in 1980 and was ranked as the forty-ninth largest U.S. corporation by *Fortune* magazine. With headquarters in Chicago, Illinois, they employ over 87,000 people worldwide; 35% of their business is outside of the U.S.

IH's several operating groups (trucks, agricultural equipment, construction equipment, etc.) have their own data processing development groups, but they share the computers operated by the corporate information systems and services (ISS) group. IH has two large data centers, in Wisconsin and Illinois, with Amdahl and IBM mainframes. Corporate ISS develops multi-group applications, such as accounting and payroll. In addition, several specialist data processing groups have emerged at headquarters. One of these is a decision support system (DSS) group, located in the corporate planning department.

The DSS group was formed by the vice president of corporate planning about two years ago. The DSS group focuses on applying management science techniques and computers to 'high leverage' corporate or operating unit problems—where 'high leverage' means a decision problem with potentially large payoffs from improving the information and analysis available to the decision-makers. The group consists of a manager, who has an operations research and consulting background, and two staff members, one of whom is a long-time IH employee with a Ph.D. in mathematics and the other is a recent graduate with a computer science background.

Shortly after its formation, the DSS group conducted a survey of the current state-of-the-art in DSS and decision support needs at IH. Index Systems of Cambridge, Massachusetts, was brought in to help with the survey. Several people at Index Systems—most notably their president, Tom Gerrity—have been involved in developing decision support systems for a number of years.

The study identified a number of areas where computerized systems would aid decision-makers and have a potentially large positive impact on the company's financial results. Among the areas identified were: fleet sales bidding, corporate financial management, production scheduling, and materials management. A prioritized list of potential projects was presented to top management and approval to proceed was given in early 1980.

Since that time, the DSS group has developed systems in a number of areas. They have found that their help is needed in several ways.

The first type of support they call 'seeding.' This involves working with managers to define their problems and information needs. The group then creates preliminary prototypes for portions of the problems to verify that the system will meet the managers' needs. Once the prototypes have been used, enhanced, and found satisfactory, the DSS team involves staff members within the operating group to create the full DSS for future operational use. The 'seeding' phase makes sure that the decision-makers are involved, that they understand what the system can and cannot do, and that the project will receive continued support.

The second type of help occurs when the DSS group operates as an independent contractor. For example, the truck group has a sales database system describing the characteristics of customers who buy fleets of trucks. Unfortunately, the database was not well suited for use in determining bid prices for large fleet orders—a high leverage decision. Truck management requested help from the DSS group to design a system that would aid in setting fleet bids for specific orders. The system was to use the existing database and other available information. After a prototype was tested, and adjusted to suit user needs, it was turned over to the truck group's information systems department for inclusion in the existing system.

In their third helping role, the DSS group acts as in-house consultants to provide assistance to management with analyses of both one-time problems as well as recurring management processes. In both cases they help the managers identify needed data, how that data can be obtained, trade-offs to be explored, and so on.

An example of a one-time DSS occurred in the area of corporate financial management. As has been noted in the public press, due to a severe six-month strike and depressed conditions in their major markets, IH has found it necessary to re-structure its corporate debt. A decision support system to assist the corporate treasury department in this area was a recent project initiated by the DSS group. Working closely with the treasury department, the group created models that simulated day-to-day operating financing requirements and costs.

These financial management models can be classified as ad hoc DSS that address cash flow problems under varying loan agreements and economic conditions. The models were created in about four weeks time using FCS-EPS, a financial analysis and modeling package from Evaluation and Planning Systems Consultants of London, England. IH uses FCS-EPS through the Comshare time-sharing service.

The financial management models were prototypes that were quickly developed to deal with a one-time decision area—negotiating the loan restructuring with IH's banks. However, they are now being incorporated in a system for monitoring treasury operations and determining optimal financing decisions under the loan agreements.

An example of a DSS to support a recurring management process is their production scheduling system. The overall project was to create a large, operational DSS for forecasting, scheduling, physical distribution, and financial analysis.

They began by creating a prototype optimization model of one plant's operations, which represented one of four components of the desired production scheduling system. Then they tested the model using eighteen months of historical data. They found that use of the model would have produced substantial cost savings—but, of course, that test assumed perfect knowledge of sales. When the test was repeated using actual sales forecasts existing during that period, they found that only about one-fourth of the projected cost savings were associated with the use of perfect information. Hence, the model could still be expected to produce substantial savings from improved decision-making with *normal* forecast error. The prototype model was refined and turned over to the operating group for implementation and production use.

The DSS group then successively worked on the three remaining components. One involved creating better forecasting models. Another was an inventory allocation model which took into account seasonal and geographic demand. And the last portion was a financial statement model for comparing the original plan to the model-produced plan.

Over the past two years, the group has drawn upon a number of 'programming' tools through time-sharing services or in-house systems. For prototyping, they use: (1) the financial planning package FCS-EPS mentioned earlier, (2) RAMIS, a data management system, from Mathematica Products Group of Princeton, New Jersey, and (3) SAS, a statistical analysis package, from SAS Institute, Inc. of Cary, North Carolina. For making management presentations, as well as for use as output from the models, they use TELL-A-GRAF, a business graphics package from ISSCO of San Diego, California. An interesting point is that all of these packages are for end user use, which is just what the IH DSS group encourages.

The major problem the DSS group has encountered is getting executives to see how the computer and decision support systems can help them, without appearing to present 'canned' solutions to replace management judgment. To overcome this barrier, the DSS group involves users heavily in projects, so the users feel they are solving their own problems, with some help from the DSS group. This approach has produced a number of successful projects and a high demand for the DSS group's services.

'Practical' decision support

We discussed decision support systems in connection with the APL programming language in our May 1976 issue. And in our August 1979 issue, we discussed tools for building executive information systems. What has happened in the area of decision support systems in the intervening time?

The answer is: quite a lot! For one thing, powerful data management systems have appeared in substantial numbers; we have discussed these DMS in many recent issues. As will be described below, DMS can be very useful for decision support. Then, too, new DSS packages and languages have entered the marketplace; these can make the building of a DSS a much easier task. Also, personal computers have arrived; as we will discuss in this report, these promise to make a large impact on managerial decision making. Finally, a growing number of managers—and even some top executives—are themselves beginning to use terminals or personal computers.

In turn, these developments have made possible what we call a 'practical DSS.' What is it? While there is no commonly-used name for the concept involved, what we mean by the term is a DSS that is quite limited in scope, is developed and put into use quickly, and helps a manager come to a decision—a decision that he might have to make on a recurring basis or one that is strictly one-time-only. Synonyms for 'practical' that we have come across in this DSS context are 'quick and dirty,' 'ad hoc,' and 'quick-hit.' To our mind, neither of the first two of these terms quite hits the mark. A practical DSS may be built quickly, but it need not be 'dirty.' And a practical DSS is not necessarily an ad hoc one; it may be used repetitively. 'Quick-hit,' used by Sprague and Carlson (Reference 4), comes closer to the mark.

A brief review is in order on some points about DSS in general. First, they provide decision *support*; they *help* the manager or executive come to a decision, they do not automatically make the decision. Secondly, they are used for decisions that are only partly structured (hence only partly computable); in each case, some amount of human judgment is needed. As an example, a DSS might be used for computing several sales forecasts under different sets of assumptions about the economy, etc. Human judgment is then needed to select the most appropriate of those forecasts.

A practical DSS can be useful for (a) getting managers started in using DSS, (b) providing decision support for certain types of management decisions on either an ad hoc or a recurring basis, (c) providing a basis for deciding whether or not to build a full DSS, and (d) for supporting decision situations where the executives cannot wait for a full DSS to be built. Also, a practical DSS can be every bit as useful for small companies as for large ones. We will discuss each of these uses in this report. In addition, we will give some caveats about the use of practical DSS.

It should be mentioned that even *Fortune 500* companies, which in general have been using DSS for years, have continual need for a good way to get new users started with DSS. Even in these companies, only a fraction of today's managers make direct use of a computer to help them with the decisions they must make. So 'getting started with DSS' is a problem confronting organizations of all sizes.

How to get a practical DSS

We talked with Steven Alter, of Consilium Associates, Inc., Atherton, California (Reference 11), about practical DSS. He made the point that a DSS can be set up quickly only in a limited number of situations. If the DSS involves mostly just the selection and listing of data from one or more data files, then a data management system can help get such a system set up quickly. If the DSS involves financial planning, then there are a number of financial planning packages on the market which can be put to use quickly. Since financial planning involves the use of some basic principles, it is quite possible that one or more of these packages will perform adequately for a DSS. Beyond these situations, it is hard to generalize about where this kind of DSS is feasible.

Within the constraints that Alter suggests, we see three main ways to obtain a practical DSS.

Reporting DSS are used for selecting, summarizing, and listing data from existing data files, to meet managers' specific information needs. Few arithmetic operations may be performed on the selected data, other than summarizing. However, if computer graphics are used (as we discussed last month), then trends, variances, etc. might be shown.

Short analysis programs can be used for *analyzing* data, not just printing it out or displaying it. And short programs can be surprisingly powerful, as we will discuss. Although not too likely to happen at this point in time, it is quite feasible for the managers themselves to write these short programs. Such programs may use only a small amount of data, which may be entered manually.

DSS 'systems' include DSS languages and other facilities, for rather quickly creating DSS to meet specific needs. A good number of DSS systems for financial forecasting are on the market, for instance; Reference 7 lists 42 of them and we know of others that are not listed.

We will discuss each of these three ways to obtain a practical DSS.

Reporting DSS

Selecting, summarizing, and listing data from existing data files is a *very* useful and practical way to support managers' decisions. In fact, it probably is—and will continue to be—the most widely used form of computerized decision support.

The new data management systems, which we emphasized during most of the past year, are excellent tools for creating this type of practical DSS. (As we have discussed, they are also very useful for supporting end user programming and for developing new applications by prototyping.)

As an example of how one of today's breed of powerful DMS can perform this role, consider the case of Tasty Baking Company.

Tasty Baking Company

Tasty Baking Company is a diversified company with headquarters in Philadelphia, Pennsylvania. Sales for 1980 were almost $180 million and they have over 2,100 employees. The major subsidiary, Tastykake, had sales of $100 million in 1980; it manufactures some four million cupcakes, individual pies, and other snack products daily, for distribution in 25 states in the U.S. Tasty Baking also has two other subsidiaries: Buckeye Biscuit Company and Phillips and Jacobs, a graphics arts supplies distributor.

In mid-1980, Tasty Baking replaced its old batch-oriented computer with a Sperry Univac 1100/60, with 2M bytes of memory, 60 terminals, and other peripherals. One of the main reasons the company selected Sperry Univac over several other competitors was the data management system that was available—MAPPER. MAPPER allows end users who have had just a few hours of training to manipulate data in a database, create ad hoc queries, and format and print out ad hoc reports, all using an English-like command language in an interactive mode. Tasty Baking believes that MAPPER will allow them to off-load some application programming onto users and thus better serve those users' needs. And they are starting to see management use MAPPER to support their decisions.

The company's graphics arts supplies distributor subsidiary, Phillips and Jacobs, is the farthest along in using MAPPER to support decision-making. This company has installed five terminals in its Philadelphia headquarters office, where they are used by the executive vice president, the operations manager, the controller, and some employees in the accounting department. Phillips and Jacobs has twelve branch offices and warehouses in the southeastern United States. The executives at headquarters use the terminals to find out the status of the inventories and sales at these locations, by making their own MAPPER inquiries into MAPPER files.

These executives attended a ten-hour class given by the Tasty Baking data processing department. At this class, they learned the ways to use more than 85 MAPPER functions, as well as how to use the system to access the appropriate MAPPER files. One of the executives, the operations manager, also attended a three-day course given at Sperry Univac where he learned how to write and store more complex MAPPER programs. This executive is the largest user of MAPPER in the company. Since he is responsible for balancing inventories with sales, he uses MAPPER to more easily access inventory and sales information and then make re-distribution and ordering decisions.

At Tasty Baking, the data processing department also has created a couple of ad hoc decision support systems for executives. In one case, the vice president of purchasing wanted to be able to project the effect of price changes of raw materials on production costs. Since this executive was not a MAPPER user at that time, a data processing staff member talked with him and then wrote the needed MAPPER program in about one day. The program uses estimated figures for sales per product to calculate the raw materials requirements and the cost to produce each product. Using MAPPER files and commands, the executive can change the price of one or more raw materials and then re-run the program to see the effect of those possible changes.

To support MAPPER users, the data processing department has a MAPPER co-ordinator. This employee was formerly a senior programmer; his new job involves guiding MAPPER users in their own 'programming' efforts. When users want to set up a new application, the co-ordinator helps them understand their data requirements, so they can create the necessary data entry screen format(s). These are created on-line and stored. MAPPER uses the data fields defined in a 'screen' to create a MAPPER file. Then the user can have the needed data entered and can use MAPPER commands to extract and manipulate the data. For the few users who write more extensive MAPPER routines, which are stored and re-run periodically, the co-ordinator reviews those programs to be sure they work efficiently. The co-ordinator also teaches the in-house MAPPER classes.

Use of MAPPER started in the middle levels of management at Tasty Baking, but the data processing manager is seeing indications that its use is spreading to upper levels of management to support decision-making.

For more information on MAPPER, see Reference 13.

Short analysis program

Yes, reporting DSS will be widely used. But suppose that the decision on which a manager desires support does not involve selecting, summarizing, and listing data from existing data files? What if not much data is involved but rather a fair number of calculations must be performed on a small amount of data? What then?

The answer here, for practical DSS situations, is small analysis programs, which John M. Nevison discusses quite well.

Nevison has written a very readable, practical book (Reference 1) on how small computers can be used to support management decisions. The book is filled with case examples plus a number of not-complex BASIC programs for analyzing the management problems.

The programs apply to such areas as: projecting costs, income, and profits, graphing sales figures, allocating fixed costs among products, project management, inventory management, depreciation calculations, and others.

A good percentage of Nevison's programs are less than 100 lines long; none are much more than twice that long. Also, most lines are remarks statements, data statements, and print statements, so the actual application logic is usually quite limited.

In his programs, Nevison uses only 16 BASIC verbs and 11 functions, plus the usual arithmetic and logical operators and the greater than, less than, and equal to relations. He contends that many managers and/or their assistants will be able to write the type of BASIC programs he has written. Nevison's programs can be used as is, or the reader is encouraged to write programs of his/her own of the same general complexity.

All of Nevison's programs can be run on a personal computer with 16K bytes of memory, he says. Most of the personal computers being sold for business uses today (such as Apple II, Radio Shack TRS-80 Model II, Xerox 820, the IBM Personal Computer, etc.) are being sold with from 48K to 64K bytes of memory. So his programs are well within the capabilities of business personal computers.

But just how practical are programs of this size and complexity—say, 80 to 100 lines of BASIC code? Are they too limited to be of much value in real-life situations? (The questions would also apply to other languages, such as COBOL or FORTRAN, but BASIC is the easiest for the novice programmer to learn.)

For an answer to these questions, consider a case example described in Lucas' new book (Reference 2). In this book, Lucas reviews salient points about successful and unsuccessful information systems that he and others have studied and have reported in the literature over the past 20 years. He defines an 'information system' as one that supports management decisions and management control; it does not necessarily include transaction processing, which makes up the bulk of routine data processing.

Perhaps the most interesting case example in Lucas' book involves a major services company with offices throughout the U.S. and Europe. Lucas singles out this case as the most successful of those reported. The vice chairman of the board of directors was considering a new employee benefit program—an employee stock ownership plan (ESOP). He wanted a study made to determine the possible impact of the ESOP on the company, to answer such questions as: how many shares of company stock will be needed in ten, twenty, and thirty years to support the ESOP? What level of growth will be needed to meet these stock requirements?. So he described what he wanted—the assumptions that should be used, and the rules that should be followed for issuing ESOP stock—to the manager of the information services department.

The manager himself then wrote a BASIC program *of about 40 lines* that performed the calculations which the vice chairman wanted, and printed out the results. These results showed the impact of the ESOP over a period of 30 years—and those results had some surprises in them. The vice chairman wondered if the results were valid, so he and the manager calculated some of the results by hand, using a hand calculator. These results tallied with the computer results, so that they felt the computer results were valid.

The vice chairman presented the results to the executive committee and, partially based on this information, the ESOP was adopted.

Some of the other executives became excited about the results of this analysis, and asked if the computer program could be used to project their individual employee stock holdings for ten, twenty, and thirty years. This was done—and it aroused even more attention. At this point, it was decided to implement the system in a more formal fashion. The company treasurer became so interested that he took over the 'ownership' of the system, and gradually expanded it to cover the planning, monitoring, and control of the various employee benefit programs.

So back to the question: how practical are BASIC programs of (say) 100 lines of code? Can they be used to support real-life decision areas? In the example Lucas describes, a 40-line BASIC program was adequate for the initial evaluation of the ESOP plan. Eventually, of course, the programs for this system became much larger in size. But the 40-line program started everything. One might reasonably say, from Lucas' case example, that programs the size of Nevison's can be useful for supporting some real-life decision areas.

One can draw another important conclusion from Lucas' case example, we think: namely, the company followed the prototype approach, which we discussed in some depth in our September 1981 issue. The area that was selected was well defined, and the vice chairman was able to state the assumptions to be used in the model. The initial model itself was simple, and dealt only with the heart of the problem. This initial model was verified, by checking some of the results by hand calculation. Then the model's results were used by the company to adopt the ESOP program. Next the model was expanded, in order to perform some other desired functions. Finally, at the appropriate point, the model was turned over to the information services department for a more formal implementation.

We found two other books that present relatively short BASIC programs for decision support purposes. Bonini, in *Computer Models for Decision Analysis* (Reference 8) describes nine real-life decision models and gives the programs for those models. Alonso, in *Simple, BASIC Programs for Business Applications* (Reference 9), gives program listings and sample outputs for 54 applications such as handling time series data, correlations, depreciation, etc.

Yes, it appears to us that the combination of personal computers and relatively short programs (often in BASIC) will play an increasingly important role in practical DSS.

DSS 'systems'

'Decision support systems *systems*' is not a particularly appealing term, but some suppliers seem to like it. What the suppliers are trying to say with this term is that their products are more than specific DSS packages or DSS languages. Rather, these products include languages, interfaces, and other facilities that aid in setting up specific DSS.

Sprague and Carlson (Reference 4) point out that a DSS *system* can be used to develop a DSS *generator*, from which specific DSS can be obtained within a class of decision support applications.

Keen (Reference 5a) describes how some 50 using organizations have attempted to cost-justify their DSS, based on their experiences with eight different DSS systems. He concluded from these experiences that DSS should be *value* justified, *not* displaced-cost justified. The benefits of a DSS are largely qualitative, he says, and most DSS evolve as users ask that more and more functions be added.

Twelve types of benefits are identified by Keen, based on the users' responses. These included: *fast response to unexpected situation* ("The model was revised in 20 minutes, by adding risk analysis; it led to a reversal of a major decision made just one hour earlier"), *ability to carry out ad hoc analyses* ("Now I can do quick-and-dirties"), *increase in the number of alternatives examined* ("8 detailed solutions generated versus 1 in previous study"), plus others.

He cited a survey of 42 organizations that were using the IFPS financial planning system and had developed some 300 applications with it. The average decision model took 5 days to build and contained 360 lines of IFPS code. (He also mentions studies of APL models, which took an average of 3 weeks to develop the prototype plus 3 to 4 months more to develop the full system.) In two-thirds of the IFPS cases, an analyst simply responded to a manager's request, and got something up and running quickly. In three-fourths of the cases, the applications replaced manual procedures. And, says Keen, since most of these companies are in the *Fortune 100* list, this indicates the limited degree to which managers in planning functions make direct use of computers.

So it appears that there are at least three ways to get practical DSS—via a reporting DSS, via short analysis programs, or by DSS systems. But practical DSS do not represent 'magic formulas.' There are some definite limitations and some precautionary measures to take.

But be careful...

For what types of situations are practical DSS suitable? Our search for information on this question uncovered the following opinions.

Alter, in a paper published in *DSS 81 Transactions* (Reference 3) gives the following guidelines for selecting a practical DSS project (he uses the term 'quick and dirty').

Clearcut goals—the goals of the project should be both settable and set at the outset; no research should be needed to define them.

Clearcut procedures—the DSS should be based on existing types of well-understood procedures and calculations; again, no research should be needed to define them.

Available data—the needed data should be readily available.

Few users—the DSS should be for the benefit of one or a few highly motivated users with common goals and concerns. The DSS should not cross organizational boundaries, nor should major selling or educational efforts be required.

Independent system—although the DSS may use input data that has been prepared by other systems, it should operate independently of all other systems once that data has been received.

While these may seem like rather severe constraints, they do point out the boundaries where DSS projects start to become large and complex.

To more clearly define the boundaries, as mentioned earlier, we interviewed Alter (Reference 11) and asked him where these 'quick and dirty' DSS would *not* be appropriate. Do not expect a *forecasting* DSS to be easy, he said. A forecasting package may or may not produce viable results in any given case. The manager must be knowledgeable about forecasting methods and must understand the area being studied, the assumptions that are made in the model, whether relationships between variables that have held in the past will apply in the future, and what variances really exist in the data.

Another area of potential difficulty is in the *allocation of resources*, said Alter. Again, sophisticated techniques are available, such as linear programming, but they usually cannot be applied in cookbook fashion. The practitioner may require a mathematics background in order to understand some of the inherent assumptions in the model being used.

Expanding on this last point of Alter's, a not-uncommon problem is for users to blindly use sophisticated mathematical techniques in a DSS. That is, the users do not understand just what the techniques do, or the assumptions on which they are based, or their limitations. This is particularly true of statistical techniques, such as regression analysis, correlation, etc. If the user unknowingly uses data that violates some assumptions of a technique, the results will be invalid.

Hackathorn and Keen (Reference 6) point out a very real problem that can arise with the practical-type DSS that we are discussing—what they call a personal support DSS. The problem is that one person's DSS can encourage decisions that are at odds with decisions of others in the same company—that is, it is not truly independent. The authors cite the case where a DSS system was used in one company to aid individual brand managers to develop their marketing plans. But the plans had to be integrated at the next higher management level. The problem that showed up during this integration was that one manager's plans would gain sales at the expense of another of the company's products. So the various managers' DSS should have been integrated with the overall organization's needs in mind, the authors concluded.

We asked Tom Gerrity, president of Index Systems, Inc. (Reference 12) what approach a user should take who is interested in practical DSS. Here are some of the points he made to us.

Start small and evolve. At the outset, select only projects where the chance of success is very high. Then start with a small prototype, if possible. The life of these initial systems may be short, but that is not important. Some users have re-written small, prototype DSS more than a dozen times, but they received far more benefits than originally envisioned.

Study the decision area carefully. Make sure you understand what is needed. (One suggestion that we have come across is: To measure your understanding, can you—using a hand calculator—calculate a few sample outputs of the kind you want from the DSS? Until you are sure you can do this, you do not understand the decision area.) Do your homework; study the available DSS packages that apply to the decision area, to see what you can learn from their use by others.

Focus on the end users. Have them *build* as much of the prototype as is practical.

Keep the DSS simple. Develop a small system that can supply a small amount of useful information on a timely basis. Later, as the users learn from the initial DSS, more sophisticated and powerful applications will evolve very naturally.

Who builds the DSS? Keen (Reference 5a) makes the point that most DSS are built outside of data processing, generally by people who know the application area well.

In the study of 300 IFPS applications, says Keen, only 3% were developed by data processing; 53% were built by staff analysts, 22% by middle management, and 22% by top management (but only 9% of the top managers actually used the terminals themselves). Our interpretation of these figures is that they indicate the relative amount of time spent on the project and do not necessarily indicate that the managers themselves wrote the IFPS code. But it is interesting that higher levels of management were deeply involved with the development of the decision models.

Rockart and Treacy (Reference 5b) say that their recent research indicates an emerging trend toward increased terminal use in executive suites. They have studied 16 companies in which at least one of the three top officers, most usually the CEO, accesses and uses computer-based information on a regular basis.

How to develop a DSS? The approach that Keen recommends, based on his studies, is a prototyping one. A 'Version 0' of the DSS is built first. Usually it is small in scale, has a limited functional capability, but is still a complete DSS, able to produce results. Based on the user's experience with this first version, the benefits are assessed and the cost of developing a more complete version is determined. Then, if it is considered worth the effort, the full 'base system' is developed.

Sprague and Carlson (Reference 4) discuss three approaches to DSS development. The first is what they call a 'quick hit.' It involves building a one-shot DSS which has no likely carry-over to the next specific DSS that will be desired. The second approach (the one they seem to favor) they call 'staged development.' It involves more initial study of the problem area and laying out a plan for an evolving series of DSS, leading to a DSS generator (from which specific DSS can be generated). Each DSS is then built with a view of carrying some of it over to the next DSS. The third approach, which is the most risky, they say, is to aim at the complete DSS at the outset.

So, no matter whether you choose to build a reporting DSS, a small analysis program, or use a DSS system, start small—build practical DSS.

Support for DSS

What type and how much help will users need for defining and building DSS? The answer seems to depend on the type of DSS involved.

Reporting DSS. A basic requirement for anyone who specifies, builds, or uses a reporting DSS is an understanding of the data definitions involved. In most cases, of course, the specifiers and users will be dealing with the application areas they know and will understand the subtle meanings of the data fields. In addition to know-ing the data definitions, it is necessary for the users to have some training in the use of the data management system, with which data will be selected and extracted from the data files. With today's DMS, this training might take from less than one hour to a few hours.

As the user expands his/her horizon and starts to extract data from a number of data files, increased support is needed. Again, it is necessary for the user to *understand* the data definitions for each file accessed. Just because two data fields in two files have the same name does not mean that their values are comparable. So someone must be available to help such users, to make sure they know when troubles can arise and how to avoid those troubles.

Then, too, such users may need programming help to select and extract data from files that are not stored under the data management system, such as tape files. Or it may be necessary to convert data from outside sources into the format needed by the data management system.

From this discussion, it would appear that the necessary support for the reporting-type DSS can be provided by data processing system analysts and programmers. And this seems to be the case, from the reports we have received. This is just one more reason to suspect that reporting DSS will be the most widely used of the three types discussed in this report.

Small analysis programs. While most managers, we gather, have little or no current interest in learning to program, it is quite possible that a good number will change their views on this once they get a personal computer. And the language they are most likely to learn is BASIC. If a manager has learned to program in BASIC, then he/she may well do the complete job of specifying, building, and using a simple analysis program.

But as these users want to expand their DSS to do more sophisticated functions, they probably will need help. At first, this may be mainly programming help, for doing things they as yet have not learned how to do. But sooner or later, the type of help needed will be in decision analysis and the selection of mathematical techniques. At this point, the services of a DSS professional will be required. So while data processing can provide some of the support needed for this type of practical DSS, professional DSS support will also be necessary.

DSS systems. Some DSS systems have been designed to be user-friendly, so that they can be used by managers and staff who have no formal computer training. But these users would require training, of course, in the use of the particular DSS system if they expected to develop their own DSS.

These are such powerful systems, however, that we suspect that most managers will soon want to exploit their capabilities. At this point, the services of a professional DSS staff (one or more persons) clearly is in order. The support needed will not be so much a case of understanding data definitions or programming techniques as it will be to study the decision areas and determine how to tackle them.

Sprague and Carlson address the question of a DSS group. Members of the group can be drawn from one or more of the following types of people: data processing application system analysts, management science or operations research people, planning department people, or staff analysts from market research, budgeting, or other such functions. In larger organizations, it is likely that DSS groups will be multi-disciplinary, with representatives from most of these functions.

All in all, practical DSS are, we think, (a) the best way for novice users to get started using computerized decision support, and (b) the best way to begin on almost *any* DSS project. That is, build a practical DSS as a prototype, use it, and decide if it is worth the effort to develop a more comprehensive DSS.

The stage is finally set, it appears, for the acceptance of computerized decision support methods by managers and executives.

REFERENCES

1. Nevison, John M., *Executive Computing: How to get it done on your own*, Addison-Wesley Publishing Co. (Reading, Mass. 01867), 1980; price $8.95.
2. Lucas, H. C. Jr., *Implementation: The key to successful information systems*, Columbia University Press (562 West 113th Street, New York, N.Y. 10025), 1981; price $30.00.

3. Young, D., and P. G. W. Keen (Eds.), *DSS-81 Transactions*, Execucom Systems Corp. (P.O. Box 9758, Austin, Texas 78766), 1981; price $25.00.
4. Sprague, R. H. Jr., and E. D. Carlson, *Building effective decision support systems*, Prentice-Hall, Inc. (Englewood Cliffs, N.J. 07632), 1982; price $24.95.
5. Center for Information Systems Research, Sloan School, M.I.T., 50 Memorial Drive, Cambridge, Mass. 02139; price $2 per report:
   a) Keen, P. G. W., "Value Analysis: Justifying decision support systems," CISR Report No. 64, October 1980.
   b) Rockart, J. F. and M. E. Treacy, "The CEO goes on-line," CISR Report No. 67, April 1981.
6. Hackathorn, R. D. and P. G. W. Keen, "Organizational strategies for personal computing in decision support systems," *MIS Quarterly* (MIS Research Center, University of Minnesota, 269 19th Avenue South, Minneapolis, Minn. 55455), September 1981, pp. 21-27; price $8.00.
7. *ICP Interface: Administrative and Accounting*, International Computer Programs, Inc. (9000 Keystone Crossing, Indianapolis, Ind. 46240), Autumn 1981, p. 24-29.
8. Bonini, C. P., *Computer models for decision analysis*, Scientific Press (670 Gilman St., Palo Alto, Calif. 94301), plus Instructor's Manual which includes some programs; 1980; price $25.00 for the two.
9. Alonso, J.R.F., *SIMPLE: BASIC programs for business applications*, Prentice-Hall, Inc. (Englewood Cliffs, N.J. 07632), 1981; price $15.95.
10. The address of Applied Economic Analysis Corporation is 4005 Locust Ave., Long Beach, California 90807.
11. The address of Dr. Steven L. Alter is: Consilium Associates, Inc., No. 4 Palo Alto Square, Suite 270, Palo Alto, Calif. 94304.
12. The address of Thomas P. Gerrity is: Index Systems, Inc:, 5 Cambridge Center, Cambridge, Mass. 02142.
13. For more information on MAPPER, write Sperry Univac, Blue Bell, Pennsylvania 19424.

With the increased costs of travel, many companies are looking to the use of telecommunications to reduce business travel. One method that receives much of the attention is full motion video conferencing. But there are numerous other, and less expensive, methods, some of which make use of computers. Data processing management should be particularly aware of developments in tele-conferencing and tele-commuting (working at or near home via tele-communications) because much of the work of data processing departments is amenable to these developments, as we discuss next month.

COMMENTARY

MICRO-COMPUTERS AT SOUTHERN CALIFORNIA GAS COMPANY

One of the questions we hear data processing managers asking these days is: "How can we encourage *controlled* use of micro-computers by end users?" We visited the Southern California Gas Company and talked with the manager of computer systems services, who told us about his company's approach to this question.

The Southern California Gas Company is the largest natural gas distribution company in the U.S., in numbers of customers. Headquartered in Los Angeles, California, they serve 3.8 million customers. They had revenues over $3.25 billion in 1981 and have 8,900 employees.

Policies regarding micros. Several years ago management in the information systems department (ISD) became concerned about the prospect of incompatible micros showing up all around the company. So they started a program to encourage controlled growth of micros. They began by issuing a company policy stating that any data processing hardware (except hand-held calculators), software, or service must first be approved by ISD.

Second, the department decided that they wanted the computers used in their office automation pilot projects and these micro-computers to eventually be able to communicate with each other as well as with their two mainframe IBM computers. So for their office automation pilots, they chose Datapoint equipment, and its ARC local area network. For their micro-computers, they recommended Radio Shack TRS-80 Model II or Datapoint micro-computers. Both will be able to operate on the ARC network, and both can emulate IBM 3270 terminals to communicate with the IBM mainframes.

To date, ISD has authorized the purchase of 24 TRS-80s. ISD requires that a requesting user describe what application(s) he expects to put on the computer, what benefits he expects, and how much money he expects the company will save from that use of the micro. ISD wants these micros to pay for themselves in a relatively short time through real savings. And, in fact, this appears to be the case. User estimates of savings have ranged from $10,000 to $60,000 a year.

ISD currently encourages only stand-alone applications for the micros—ones that will not need to interact with the mainframes. Also, applications currently on the mainframes which do not interface with other applications can be transferred to the micros—and have been, in several cases. ISD expects the users to be responsible for their own data and for their own application development as well.

Generally ISD recommends a TRS-80 Model II with 64K bytes of memory, an additional dual floppy disk drive (for a total of 1.5M bytes), and a daisy wheel printer. The system costs about $9000 and is bought under the office equipment account of the administrative services department. The software that is recommended includes: the TRSDOS operating system; VisiCalc, the electronic spreadsheet package; BASIC; and two Radio Shack packages—Profile, a file management and report generator package and Scripsit, a word processing package. The packages used most often by the users are VisiCalc and Profile; BASIC is used occasionally.

Uses of the micros. Some of the users use VisiCalc for budgeting and cost analysis or Profile for generating reports. Here are a few more unusual uses of the micros.

The transmission department maintains the pipelines and under-ground storage facilities that supply natural gas to the company's distribution system. The company has some 120 billion cubic feet of natural gas stored in under-ground sites, generally depleted oil fields. These sites require maintenance; in particular, they require drilling of new wells to make movement of gas into and out of the storage sites more efficient.

The transmission department uses its TRS-80 to keep track of well-drilling contractor information—such information as which contractors specialize in drilling in certain types of earth and rock, past contract work, how efficiently they drill, and so on. This system was created by an engineer who knew BASIC.

In the claims department, the TRS-80 has been used to replace a mainframe application. There are occasions when claims are filed by the gas company against a contractor, a company, or an individual (and vice versa). For example, if a contractor is digging and hits a gas main causing damage, the company may file a claim. The claims department has created a claims file so that they can compare frequencies and types of claims. Using this information they can better determine how each claim should be processed. The word processing package is used to create the letters and envelopes for the claims correspondence.

The industrial engineering department, which is a part of ISD, maintains engineered work standards for the company. Last year the department wanted to change the way that the field services standards would be used by the distribution divisions. The company has many field crews that repair pipes, install new pipelines, and so on. The work done by these crews is recorded and actual performance is compared to the standards.

To perform this actual-versus-standard monitoring and reporting, the industrial engineering department recommended that the distribution divisions put the application on TRS-80s. The divisions concurred, but felt that since this application would probably be used by all 13 distribution divisions, it should be developed by ISD. After much discussion, the users agreed to write the application. They created a task force, which included one person from ISD, to design and oversee the project. The application was created using VisiCalc and it is now being tested in two divisions.

In all, ISD is pleased with their controlled approach to introducing microcomputers to their end users. And they feel they will eventually be able to offer a powerful office automation network to further support these users.

SUBJECTS COVERED BY EDP ANALYZER IN PRIOR YEARS

Coverage code:

A	Office automation
B	Using minis & micros
C	Managerial uses of computers
D	Computer message systems
E	Distributed systems
F	Data communications
G	Data management and database
H	Analysis, design, programming
I	Computer operations
J	Personnel
K	Introducing new technology
L	Security, privacy, integrity
M	New application areas

(List of subjects prior to 1979 sent upon request)

Prices: For a one-year subscription, the U.S. price is $60. For Canada and Mexico, the price is $60 *in U.S. dollars,* for surface delivery, and $67 for air mail delivery. For all other countries, the price is $72, including AIR MAIL delivery.

Back issue prices: $7 per copy for the U.S., Canada, and Mexico; $8 per copy for all other countries. Back issues are sent via AIR MAIL. Because of the continuing demand, most back issues are available.

Reduced prices are in effect for multiple copy subscriptions, multiple year subscriptions, and for larger quantities of a back issue. Write for details.

Please include payment with order. For U.S. subscribers, you can use your Visa or MasterCard charge card; include your card name, number, and card expiration date on your order.

For payments from outside the U.S., in order to obtain the above prices, take your choice of three options: (1) use an international money order, (2) pay in U.S. dollars with a check drawn on a bank in the U.S., or (3) use any of the following charge cards: Visa, MasterCard, Eurocard, Access Card, Standard Bank/Kaart, Union Card International, or Diamond Card International. Please be sure to include your card name, number, and card expiration date on your order.

Editorial: Richard G. Canning, Editor and Publisher; Barbara McNurlin, Associate Editor. While the contents of this report are based on the best information available to us, we cannot guarantee them.

Missing Issues: Please report the non-receipt of an issue within one month of normal receiving date; missing issues requested after this time will be supplied at the regular back-issue price.

Address: Canning Publications, Inc., 925 Anza Avenue, Vista, California 92083. Phone: (714) 724-3233, 724-5900.

Microfilm: EDP Analyzer is available in microform, from University Microfilms International, Dept. P.R., (1) 300 North Zeeb Road, Ann Arbor, Mich. 48106, or (2) 30-32 Mortimer Street, London WIN 7RA, U.K.