that it should never execute, it makes a call to a routine DISABLE, which bails out in a graceful fashion. All these errors cause the controller to print the message:

FATAL ERROR IN MODULE
MODULE DISABLED

and to refuse to allow the user further access to that module, a severe measure designed to prevent incorrect answers and to insure that the problem gets reported.

Surprisingly, only 18 percent of the disablings were ever communicated to the system designers. The logging file was the only record that most of these failures occurred. Later contacts convinced us that the typical user feels the error is somehow his fault. He doesn't wish to report the problem lest it reflect upon him or his understanding of the documentation, the syntax or the analysis technique itself. He suffers from a basic belief that the computer is more likely to be correct than he is.

The logging file was also useful for determining the average cost of each command and the most frequently used techniques. In one instance a command was found to be very costly for certain data sets due to a paging problem. This was corrected by rearranging the working set. Frequency of use was a guide to making efficiency improvements and installing new features. For example, improvements in the efficiency of TRANSFORM yielded benefits for all users. Regression, cross-tabulation and analysis of variance occupied the bulk of the usage.

Conclusion

A logging file combined with error detection tools is an important means of measuring the success of a DSS, as well as maintaining and improving it. Users cannot be relied upon to perform this function. Successful results with TACTICS suggest the importance of using the simplest feasible syntax.

AN APPROACH TO AN ADMINISTRATIVE DSS

By M. Loomis, J.F. Nunamaker
and B.R. Konsynski
Management Information Systems
University of Arizona

Implementing a decision support system which builds on existing data files in an established environment can be effectively accomplished by a hybrid "bottom-up with top-down control" methodology. The existing data are the basis for the bottom-up aspect of the approach, which addresses implementation; the users' ultimate requirements are the basis for the top-down aspect of the approach, which addresses design and system structure. This approach contrasts both with a pure top-down approach requiring the establishment of new data structures and with a strictly bottom-up approach which creates coordination problems.

The authors have successfully applied the methodology in development and implementation of an integrated decision support system providing information for operational and budget planning at the University of Arizona.

In the bottom-up with top-down control approach, the decision support system development project commences with iteratively refined specification of outputs, processing and required inputs. Users and analysts meet to establish system objectives and user requirements, including information needs, data sources, integrity and validation specifications. The major subsystems of the decision support system are identified.

With top-down control factors established by a complete and consistent statement of logical requirements, the analysts can proceed with bottom-up elements of the implementation, resulting in an interim system consistent with the logical design resulting from the top-down control effort and directly interfacing with existing data and operating systems. The decision support system can thus become operational with minimal effort, yet in the longer term the system can evolve according to the top-down design structure.

Systems to support policy planning and decision making by higher management in an organization are fundamentally different from systems to support daily operations, transaction processing and computer hardware. Success of a planning system is generally much more dependent upon human factors in the organization. One implication of the bottom-up with top-down control methodology is that an organization should develop its own decision support system, rather than purchasing a system developed for other institutions.

Concerning aspects of the methodology, the approach calls first for the active and informed participation in the design process by the eventual users of the system. Thus pitfalls of applying inappropriate assumptions and implications from another environment can be avoided. An information system that successfully serves the needs of a complex and socially interactive policy planning system must be tuned to the decision-making patterns of the participants. Participation of users in the design effort results not only in an improved design, but can materially reduce the education and selling effort required for acceptance of the system.

Second, all current files that could impact or relate to the planning system are analyzed. Thus a comprehensive file structure supporting the entire information and planning system and oriented toward an integrated data base can result. In general, implementation of an externally created system would introduce a layer of files in addition to the existing ones, potentially complicating problems of compatibility and extension.

Third, the major problem of data quality is directly addressed in the advocated methodology; this problem is more difficult to resolve during implementation of an imported system. Any system will fail if the data supplied to it are incorrect, incomplete, untimely and so on. Additionally, a system developed without knowledge of current operating procedures is likely to impose new demands on operating people with probable concomitant decline in data quality.

A systems design tool, PSL/PSA [1] has proved valuable in documentation of data items and their sources, then in analysis of their similarities. Assessment of data quality can reveal basic data problems: inconsistent redundant data, lack of relationship or communication between data collectors and users of those data, lack of correlation of coding schemes across existing files, inadequate standards for file definition, file structure volatility, poor documentation of data sources, users and responsible parties, inadequate training and operating manuals and so on. Existing features which are potential obstacles to quality can be redesigned. Finding data inconsistencies later in the project (had a purely top-down approach been pursued and eventually reached the bottom: basic data) would have significantly delayed generation of meaningful reports.

Fourth, an interim decision support system which generates useful reports from the available data is implemented. Because the bottom-up phase of development is always controlled by the top-down overall system view, the analysts are not apt to make decisions at this stage of implementation which would later prove incompatible with a complete system data base. Anomalies detected in the early reports are used to identify data problems. Because available data are used as early as possible, problem areas are found much sooner than if reports had resulted from a purely top-down approach.

Fifth, because the overall system design guides the bottom-up implementation, the resultant software will be easily convertible to data base access in the future. The bottom-up approach alone can not promise this potential convertibility.

Using the bottom-up with top-down control methodology, the authors were able to produce useful reports for users much earlier than had a purely top-down approach been followed. Several months after project inception, the users received the first set of instructional summary reports; four months later they were also receiving space and budget summary reports. It is estimated that no reports would have been available until several more months after project inception had an outside system been purchased.

Had a top-down approach been adhered to, the delays would have been even greater. A purely top-down approach is likely to have a backtrack (which then makes the approach interactive, from later stages, rather than top-down anyway) as data are found to be unavailable and requirements have to be refined to make implementation feasible, meanwhile incorporating significant delays before the system can begin to support the decision making process.

Reference

Nunamaker, J.F. and Konsynski, B.R., "From Problem Statement to Code Generation," *Systemeering 75,* Student Literatur, Lund, Sweden, 1975.

MODELING UNSTRUCTURED DECISION MAKING

By Richard D. Hackathorn
Department of Decision Sciences
The Wharton School
University of Pennsylvania

Introduction

This paper summarizes a new approach for modeling the decision-making activity within an organizational context.* The approach, called activity analysis, is a micro-level, discrete process analysis of the activity associated with the information system in the organization. One purpose of the analysis is to delineate unstructured decision-making activity, thereby assisting in the design of decision support systems.

In the information-rich world in which organizations must operate,[14] the capacity of persons to make wise decisions is largely a function of how information is managed within the organization. The information system (the mechanisms that manage the information within the organization) should be "centered around the important decisions of the organization, many of which are relatively unstructured."[3] A decision support system (DSS) is that portion of the information system concerned with supporting unstructured decisions.

The reason that most existing information systems have not adequately dealt with unstructured decision making can be attributed to a lack of understanding about decision processes, especially within a specific organizational setting. By definition, unstructured decision making implies that there is an absence of a routine procedure within the various decision phases—problem definition, alternative generation, choice selection and so on. The relevant models of a decision-making situation are "often the unverbalized models used by the managers of the organization."[3] The key ingredient is "the skill to elicit from management its view of the organization and its environment and to formalize models of this view."[3] In general, there are two ways of eliciting these views: ask for the

**This paper is based on the author's dissertation research[4] and is a summary of another paper.[5]*