1.0 DECISION SUPPORT SYSTEMS--AN OVERVIEW

Traditionally, business organizations have employed their computers in the area of operations support—for recordkeeping, transaction processing, and generating periodic routine reports. Today’s managers and professionals, however, are demanding hands-on access to data and computerized systems that provide solutions to a variety of “ad hoc” problems, many of which have never before been a part of the data processing (DP) mainstream.

The Problems

As a result of these heavy demands, several well-known problems have surfaced:

- **Application Backlog** - DP departments have become severely backlogged with requests for information and application development. This includes the invisible backlog of requests that decision-makers do not even consider making due to the growing documented backlog.

- **Complexity** - Ad hoc requests often call for the integration of data from several functional areas and are therefore more complex than production systems and report oriented MIS’s.

- **Knowledge of the Problem** - The person who best understands the needs and requirements of the problem is usually not in DP, but is the decision-maker who made the original request. Unfortunately, this individual often has little understanding of how to use the computer to process his request.

- **Rapid Response** - In many cases, the request for “strategic” information must be supplied quickly in order to be useful. Any of the above problems may delay the processing of these important requests.

The Solution

The general solution to these problems is well-known also: implement end-user computing. But the difficulty is in implementing it in a manner that will preserve the integrity of the organization’s information resource—the production data base—while at the same time providing end-users with easy to use, flexible query and powerful data analysis capabilities.

**Short Term Success**

In an effort to place more computing power directly in the user’s hands, many organizations have implemented Information Centers and Decision Support Groups that are separate from their application development centers. These groups work directly with users and help the user in developing their own computer applications. The use of microcomputers for end-user computing is another alternative. This strategy is often implemented in conjunction with an Information Center or Decision Support Group.

Initially, these approaches often meet with a high level of success. The keys to this success lie in their use of:

- **Nonprocedural application languages** - Computer languages that require the user to tell what is to be done but not how to do it.

- **Interactive computing** - When analyzing and interacting with the data in an information analysis mode, the user needs quick response from the computer to maximize the creation effort.

- **Graphic presentation** - The presentation of data through graphics is a valuable aid in transforming “raw” data into useful information.

**Long Term Success**

This type of end-user computing, however, is unlikely to succeed in the long run unless a structured architecture is employed in developing the end-user applications. Although they employ high level tools such as fourth generation languages, Information Centers operating without a structured architecture will lack the capacity to fulfill and maintain a large number of requests and fall prey to the same problems that the DP and MIS groups have encountered.

Microcomputers, although multifunctional and relatively inexpensive, also have serious long term shortcomings. Today, downloading data from and uploading data to the corporate data base with the micro requires technical support and supervision. Over time, this problem will be solved by technology. More importantly, applications and local data bases are often developed independently by each department. Consequently, coordination and control of the data becomes extremely difficult, the integrity of the data suspect, and the reliability of the inferences drawn from the data is low.
These long term problems can be resolved by selecting the appropriate decision support tools and employing them within a structured design and implementation methodology. In this manner, Decision Support Systems (DSS) facilitate effective decision-making by providing an organized, structured interface between the decision-maker and the computer. By shifting daily operation of the system to the decision-maker, the person with the most knowledge about the application can access the data to create information. This permits "what if" questions to be addressed as they arise in the course of normal business operations. Moreover, since they no longer act as intermediaries for every request, information center personnel may concentrate on the issues of coordination and integrity of applications that require their expertise.

A complete, effective DSS has the following capabilities:

- System generation and migration;
- System prototyping;
- System maintenance;
- Data base management;
- Interfaces to organizational data;
- Nonprocedural query and analysis;
- Powerful data analysis and graphics;
- Microcomputer interface.

In this paper, we describe the design methodology and system architecture used to develop a set of integrated SAS-based decision support tools (DS/Software). We conclude by describing an application DSS built from these concepts, the Bid Analysis and Management System (BAMS).

2.0 DS/ DESIGN METHODOLOGY AND SYSTEM ARCHITECTURE

In order for a DSS to exist, it must be constructed within the context of a particular application. Without the application, the DSS software is merely a set of tools in search of use. To be effective, the decision-maker must understand what the DSS can do and the system developers must understand how to integrate DSS technologies into the decision-making process. For this to occur, an organized, structured design methodology for building the DSS must exist. The methodology for development is shaped by the components of the DSS environment:

- a generalized system architecture,
- the system development tools, and
- the user tools.

Methodology

With the proper design methodology, an organization can establish the framework for building a structured DSS from its unstructured perceived needs. The design methodology includes:

- Entity analysis - Identifying the information required to support a decision-making process and relating this information to "real world" objects about which an organization may want to collect data.
- Data base structuring - Using relational data base design methods to define the relations to be included in the DSS data base. The DSS data base reflects the entities for the DSS.
- Model building techniques - Defining the information processing models that will create the desired information from the data in the data base.
- Heuristic refinement and prototyping techniques - Building a prototype of the data base and the models and allowing for timely evolution that is a reaction to changing needs.

System Architecture

Key to the methodology is the concept of a DSS architecture. Adoption of a generalized system architecture gives a common structure and organization to each DSS built using the architecture. This is important for maintenance of a large number of systems, the training of system users, automated system generation, and the development of analysis expertise. The DSS architecture provides modular components and flexibility to support a variety of systems and source data bases, while allowing simple interfaces between the components.

Figure 1 shows a generalized DSS architecture with its main components:

- An automated system generator;
- An information extraction system;
- A relational data base;
- A library of information analysis models;
- An interactive dialog manager;
- A microcomputer interface.

Within the architecture, each component plays an integral role.

The automated system generator consists of a set of interactive system design, definition, generation, documentation, and evolution tools which aid in the development and installation of a DSS application. Through an interactive dialog, the user creates an application data dictionary by describing the DSS data base. The system generator then (1) automatically generates the application extraction code necessary to extract data from the outside world and to update and maintain the data base, (2) assembles and organizes the information into relations to form the DSS data base, and (3) produces design and system documentation for...
models are application programs which are used for (1) the application systems can be supported with the analysis programs, e.g., graphics on-line data entry and edit, and (4) a data base for user-defined data sets produced from application analysis, an interface to permanent data base. Erroneous data is automatically rejected and placed in a temporary transaction file. The user edits the bad data interactively and the process is repeated. The architecture also provides for user-defined data sets or other output data sets.

The information extraction system maintains the permanent DSS data base via three loosely coupled components: (1) external data residing in production data bases, (2) key entered data, and (3) on-line data entered interactively. Data is extracted from external data sources, edited, and added to the permanent data base. Erroneous data is automatically rejected and placed in a temporary data set on a recurring basis and have been "defined" to allow both the inexperienced and expert user to use this system with the same degree of ease and satisfaction. Data management involves an interface to the information extraction system; data analysis, an interface to predefined and ad hoc information analysis models; and system management, an interface to environment and application maintenance functions. The architecture is designed so that multiple application systems can be supported with the same dialog manager.

Information analysis models are application programs developed to manipulate, analyze, and display the data contained in the DSS data base. Their purpose is to transform the data into useful information for the decision maker. The DSS architecture allows for two types of models: predefined models and ad hoc models. Predefined models are application programs which are used on a recurring basis and have been "defined" to the dialog manager. These models are designed with a series of special macro statements incorporated into the programming code to allow for flexibility in analysis and output. The program's parameters are specified by the user through the screens of the dialog manager. This allows the user to custom tailor a model to meet specific needs without knowledge of how to program. Ad hoc models are user-created programs maintained by the dialog manager for special purpose needs. The architecture permits ad hoc models to evolve into predefined models.

**DSS Development Tools**

Establishing a generalized system architecture is a necessary first step in building a DSS. A more formidable task may be in identifying and implementing a set of software tools which make the DSS operational. Figure 1 illustrates two obvious software needs:

- an application language that supports the processing activities associated with data analysis, management, and presentation; and
- an interactive display management language that allows the development of user-friendly dialogs.

When selecting these software tools, a few guidelines should be followed. First, the tools must significantly reduce the cost of development, maintenance, and execution of the system. Second, the tools must be available on a wide range of potential user's computing environments. For these reasons, the SAS System is recommended as the application language and primary development software and IBM's ISPF (Interactive System Product Facility) as the interactive display management language.

The SAS System fits the established criteria better than any other single application package currently available by providing the DSS with:

- a high level programming language;
- powerful data management and retrieval tools;
- a library of over 75 statistical analysis procedures;
- full-screen data entry, edit, and information processing tools; and
- color graphics.

ISPF is selected for its powerful dialog management capabilities and because it spans the IBM operating system (MVS, CMS, and VSE).

SAS and ISPF do not, however, provide all the capabilities promised by the DSS architecture of Figure 1. For example, although the SAS System offers one of the most complete set of tools for manipulating data that can be found in a software package, these tools are relatively low-level and require a procedural approach. That is, users must develop some proficiency in SAS programming in order to take full advantage of these data management tools. Conversely, it is the nonprocedural approach to data analysis and graphical presentation that is the strength of SAS in meeting the needs of the typical user. Other DSS developers have recognized the need for a more user-friendly environment in the area of data management and retrieval. However, the data analysis and application tools of these systems either require programming skills or, if user friendly, lack the power and versatility of SAS. What is required is a nonprocedural interface to SAS.
which gives the typical, unsophisticated end-user direct access to the DSS data base for data management and retrieval purposes.

In Section 3.0 we present a set of DSS user tools built using SAS and ISPF and designed to perform the functions outlined by the DSS architecture.

3.0 DSS USER TOOLS

Based on the DSS architecture, Info Tech has developed an integrated set of five decision support user tools as part of its DS/Software product line. These include:

- **DS/GEN**—an interactive dictionary-driven system generator,
- **DS/DM**—a full-screen interactive dialog manager,
- **DS/DB**—a direct-access facility for SAS data sets,
- **DS/QL**—an SQL-based query language for SAS,
- **DS/PC**—an interface between the microcomputer and the DSS data base.

Whether used together or individually, these easy-to-use tools provide support for any application requiring data management, query, analysis, and display. As a package, they form a complete, effective SAS-based DSS.

The particulars of each of these five user tools and their roles within the DSS architecture follow.

**DS/GEN -- Automated System Generation and Maintenance**

DS/GEN fulfills the role of the automated system generator described in the system architecture. With DS/GEN, the user performs three main functions -- (1) system definition, (2) system generation, and (3) system evolution -- through an ISPF-based menu-driven interactive dialog; no computer programming is required.

DS/GEN'S interactive dialog provides an interface to SAS/FSP screens so that the user can construct a comprehensive data dictionary of the SAS data base which defines the particular decision support application. At the user's request, computer-generated workbooks and design documents are automatically produced to aid in system design and analysis. Once the system is defined, DS/GEN provides facilities for automatic system generation. In addition to a fully-documented data library of SAS data sets from the data dictionary, DS/GEN creates the programming code necessary to load and maintain the DSS data base, including facilities for: data extraction from other data bases and files through the information extraction system; on-line data entry and editing through SAS/FSP; creation of data entry forms; data base update; data base security; creation of data base archives; and data base back-up/restore.

**DS/DM -- Interactive Dialog Manager**

The Interactive Dialog Manager, DS/DM, consists of a structured hierarchy of full-screen ISPF menus and panels through which various data analysis, data management, and system utilities are performed (see Figure 2). DS/DM's controlled interface to the often confusing features of MVS, TSO, JCL (or VM/CMS) makes the computing environment easily accessible to new users while at the same time dramatically reducing user-training and the need for technical support. Since no knowledge of computer programming is necessary to operate DS/DM, end-users have "hands-on" access to the DSS data base.

DS/DM interfaces to the information extraction system created by DS/GEN, allowing the user to execute programs that extract data from the outside world and load them into the SAS data base. Other data management functions include data entry and editing through a table-driven interface to SAS/FSP and data base backup/restore through an interface to SAS utility programs created by DS/GEN.

For data analysis, DS/DM interfaces to a library of information analysis models, as depicted in the DSS architecture. Novice or sophisticated users can execute predefined analysis programs, create new SAS programs, and browse the SAS data base using SAS/FSP. DS/DM stores information about the user customizable predefined analysis programs in the form of a model profile. Model profiles are structured so that they may be easily tailored to analyze and display information in response to current needs. DS/DM also provides an interface to the
naked SAS environment for users who wish to create their own SAS programs. Any reports, tables, or color graphics produced by analysis models (either predefined or user-defined) can be viewed at a terminal or the output directed to hard copy for presentation purposes.

**DS/DM** provides system utilities for definition and maintenance of the DSS interactive environment. These utilities include maintenance of system parameters, initialization of the DS/DM session for a new system, maintenance of system passwords, and maintenance of model profiles. In addition, DS/DM provides a security control feature so that other users have access to only the systems, data, and analysis programs for which they are authorized.

**DS/DB -- Data Base Facility**

Although modern data base design techniques may be used to good advantage with the SAS System, they employ relatively inefficient sequential data access methods. For a decision support environment to be truly effective in medium and large data base applications, it must be efficient in its data accessing techniques without requiring sophisticated programming on the part of the user. DS/DB is an enhanced data base facility designed to greatly increase access efficiency for SAS data sets, while decreasing overall programming effort.

DS/DB provides direct access to observations in SAS data sets through the use of two specially written SAS procedures, DBBUILD and DBSEARCH. DBBUILD creates and maintains a data base storage structure, called a data set cluster, for an existing SAS data set. The cluster consists of three data sets, an INDEX, OCCURRENCE, and DATA data set, which define the target data set and the key variables used to subset the data. DBBUILD provides facilities for updating of the data set cluster and an unlimited number of key fields. The following statements illustrate the use of DBBUILD to create a SAS data set cluster with part number, bolt size, and part name as index keys:

```
PROC DBBUILD DATA=PARTS;
  KEYS PARTNUM SIZE PNAME;
```

The DBSEARCH procedure selects observations from a SAS data set cluster using the index keys defined by DBBUILD. DBSEARCH employs optimum search techniques for direct access to the data and creates an output data set ready for analysis. For example, the statements below find the part numbers of all size 20 bolts:

```
PROC DBSEARCH DATA=PARTS OUT=RESULT
  (KEEP=PARTNUM);
PARMCARDS4;
  IF PNAME='BOLT' AND SIZE=20;
```

DS/DB can either stand alone as a pair of SAS enhancement PROC's or be integrated within the DSS architecture to provide direct access to SAS data while significantly reducing disk I/O's and CPU time.

**DS/QL -- Query Language**

The need for a nonprocedural SAS data management facility within the DSS architecture was identified in Section 2.0. DS/QL is a nonprocedural query language for SAS designed to meet the data management and retrieval needs of nonprogrammers and sophisticated computer specialists.

DS/QL incorporates most of the retrieval features and syntactic of the popular SQL language and includes many enhancements which take advantage of the SAS environment. As a consequence, data can be retrieved and analyzed using SQL within SAS programs. Powerful nesting capabilities permit the selection of observations based on variables and values across an unlimited number of data sets within the SAS data base.

Each DS/QL query can be thought of as an English-like question or statement structured in the form of SELECT...FROM WHERE. The query is then translated by DS/QL into equivalent SAS programming statements. For example, the following query will retrieve the names of Florida vendors.

```
$ SELECT NAME
  FROM VENDOR
  WHERE STATE = 'FL' ;
```

In addition to retrievals, the results of a DS/QL query can be displayed and analyzed through DS/QL Display statements. This includes listing, statistical analysis, graphics capabilities with simple display statements such as PRINT, PLOT, HISTOGRAM, and CHISQ.

Within a decision support environment, DS/QL can be accessed through the interactive dialog manager. DS/DM, and can retrieve data from SAS data set clusters created by DS/DB. By placing the powerful data retrieval and analysis features of SAS into the hands of the end-user, DS/QL improves the productivity resulting from using a SAS-based DSS.

**DS/PC -- Microcomputer Interface**

For micros to be truly useful as a decision support tool, they must be fully integrated into the decision support environment. This requires the necessary hardware and software to interface the micro to the mainframe computer. DS/PC software is a set of programs and interfaces designed for the micro end-user.

DS/PC brings the decision support environment of the mainframe to the micro through a user-friendly dialog consisting of menu screens and data entry panels. This menu-driven interface eliminates the need for users to learn confusing or complicated operating system commands. DS/PC interfaces directly to DS/DM, permitting the micro end-user to perform analyses of the mainframe DSS data base. Reports and graphical output can be downloaded to the micro and directed to hard copy at the micro workstation's printing and plotter facilities.
DS/PC also provides the end-user with the structure and control necessary to upload and download data between the micro and the mainframe. Through a new SAS PROC, called DSXPORT, data can be downloaded from the DSS data base and formatted for any of the most common microcomputer analysis systems.

4.0 A DSS APPLICATION: BAMS

Info Tech has employed the methodology and architecture of DS/ to build the Bid Analysis and Management System, BAMS. BAMS is currently being installed by a number of State Highway Agencies to assist them in contract cost estimation, the bid award process, and in detecting bid-rigging activity in the road contracting market. In addition to supporting bid analysis and review, BAMS addresses the much broader need of State Highway Agencies to have flexible and easy access to the wealth of data that is generated by the highway contracting process.

Typically, BAMS combines data from the design and estimation phase, the bid letting and award phase, and the construction and final payments phase of the highway contracting process. As a result, data must be entered from many different machine readable and manual sources. To meet the need for efficient data entry and organization, DS/GEN is applied. Through DS/GEN, the system builder first defines and organizes the BAMS data base into a library of SAS data sets according to the logical and meaningful relationship among the data elements. DS/GEN then automatically creates the extraction code necessary to load the external data into the data base through the information extraction process.

All further BAMS operations are now facilitated through DS/DM menu and screen panels. Figure 3 is an example of a Master Menu selection panel designed for BAMS. The system maintenance facilities of DS/DM allow the BAMS installation manager(s) to define the interactive DSS environment. Once the environment is established, users can perform maintenance and analysis of the BAMS data base.

The key to data analysis is a library of predefined SAS application programs called BAMS Analysis Models. BAMS Analysis Models are designed to summarize and display information on the bid award process both graphically and numerically. Typically, these models cover the following areas:

- Market shares analysis;
- Vendor activity analysis;
- Price analysis;
- Geographic summary statistics;
- Bid evaluation;
- Engineer's estimates evaluation and review.

BAMS users do not have to directly access the hard coded SAS programs. Instead, they use the DS/DM Model Profile Directory. The directory contains a set of model profiles, each one corresponding to a particular analysis program. Information stored in the profile can be customized by the BAMS user to meet his specific needs. For example, in a market shares analysis model profile, users can choose to calculate "gross" market shares or "dynamic" market shares. Selections are made via menu and data entry panels within the dialog manager. When a batch or interactive job is submitted, DS/DM generates the proper SAS MACRO code in the market shares program to create a customized version for execution. Where appropriate, the SAS code for these models can incorporate the direct access and data retrieval features of DS/DB and DS/QL, respectively.

DS/DM also provides BAMS users with the ability to create and execute ad hoc SAS programs through an interface to the SAS environment. In the SAS environment, users only need to enter only the proper SAS code for their program; no JCL commands are required. DS/QL and DS/DB statements can be used to good advantage in these ad hoc SAS programs.

In BAMS, State Highway Agencies have a user-oriented, full DSS built using the generalized system architecture outlined in Section 2.0 and designed specifically for the management, analysis, and monitoring of bidding data pertaining to the highway construction industry.

5.0 SUMMARY

This paper outlines a methodology and generalized system architecture for building a complete, effective SAS-based DSS. The main components of the DSS include:

- an automated system generator;
- an information extraction system;
- highly flexible data base management;
- sophisticated modeling capabilities;
- a user-friendly interface;
- microcomputer interface to the data base.

To fully develop the DSS, several software tools must be integrated within the system architecture. We have identified SAS and ISPF as the primary development tools, and a set of five user tools: DS/GEN, DS/DM, DS/DB, DS/QL, and DS/PC.

The decision support methodology adopted is general enough that it can support a wide variety of DSS applications, in addition to those implemented in BAMS for State Highway Agencies.

* SAS and SAS/FSP are registered trademarks of the SAS Institute Inc., Cary, NC.
** ISPF and SQL are trademarks of IBM Corp.
*** DS/DM, DS/GEN, DS/DB, DS/QL, DS/PC and BAMS are trademarks of Info Tech.
Figure 1. DSS Architecture

Figure 2. DS/DM User Interface

Figure 3. BAMS Master Menu