I. Introduction

This paper attempts to prescribe the best system (method) for generating information systems. Since the function of an information center is to encourage, train, and outfit users for their information tasks, the center and its users form a process for generating information systems. Implicitly, at least, therefore, this paper defines the best system for the information center. That system will be "intelligent", as Dr. Paul J. Werbos describes in his paper, "Generalized Information Requirements of Intelligent Decision-Making Systems", found elsewhere in the 1986 proceedings. It will be a generalized system able to maximize some measure of success over the long-term in an uncertain environment to which it must learn to adapt in an open-minded way. In this paper that is interpreted to mean the system of the information center must be capable of generating widely diverse information systems while maximizing the user's computer skills and task knowledge, minimizing formal training, and adapting easily to changes in requirements.

The paper's approach is conceptual and deductive. It examines the intrinsic method by which humans create information systems and then deduces the process which would best facilitate that natural method of information system design. Its scope is general and holistic. The guidelines prescribed are applicable to any information center, regardless of the languages, utilities, or hardware it employs. Furthermore, the system of the information center is seen as an inseparable integral of people, tools, procedures and purpose and thus the paper's prescriptions reference all of those elements.

Readers who wish more information on the natural way humans create information systems should read the Herbert Simon articles cited in the bibliography. Those seeking information on specific applications embodying at least some of the prescriptions of this paper should examine the articles referenced by brackets [ ] and also read the following papers of the 1986 proceedings: "User-Friendly Systems - 'Easy to Learn' Vs. 'Easy to Use"', Michael H. Callahan; "The Use of the SAS Macro Facility in the Development of a Clinical Information System", Thomas R. Hoffman; "Using SAS/PSP® and the Macro Facility to Create a Generic Report Writer", J. Page Miller.

II. What Is An Information System?

The method by which appropriate data is gathered, verified, manipulated, presented, and evaluated; i.e., the process by which information is created.

III. What's Wrong With Most Information Systems?

A. They take too long to create.

B. They don't fulfill the user's purpose.

C. They're too difficult to learn or else quickly become restrictive and time-consuming in their operation.

IV. Why Do Those Problems Exist?

A. Because the process used to create the system is inappropriate. It's meant for systems for which the user's requirements can be specified in fine detail before design and coding begins [4, Pp. 45-54]. Such systems are very structured. They usually involve a single subject, such as inventory, which provides a clear though, often, narrow focus and a common terminology.

B. Because the functions selected for the system are inapplicable or inadequate.

1. Are too limited in the data they can manipulate.

2. Can't perform the right operations.

3. Can't be organized in the appropriate sequence.

4. Can't be dissected and reassembled to produce new functions.

5. Are tied to one or two application software languages.

6. Are too oriented to report generation instead of data creation for further manipulation.

7. Don't have a short-hand form of reference for the users who have become skilled in their application.

C. Because the right personnel aren't supporting system development

They

1. Don't enjoy teaching
2. Don't communicate well  
3. Don't have enough curiosity  
4. Don't understand the need to sell the system or else don't want to.

V. What Can Be Done to Improve Information Systems?

A. Understand problem-solving and its relationship to the development of an information system [1,2].

1. Simple problem solving can be described as uniformed search. It has the following elements:
   a. A starting point (knowledge state) is identified.
   b. A generative process (operator or function) is applied to the input data representing the knowledge state to produce a new knowledge state (output).
   c. A test (evaluation) process is applied to the knowledge state to determine whether it represents the desired or best knowledge, i.e., whether it is the solution.
   d. A decision is made to continue or terminate problem solving. If the decision is to continue, the problem solver selects at random a different starting point or generative process.

2. More complex (heuristic) problem solving can be described as informed search. The problem solver gains information from one cycle of the problem solving which aids his selection of knowledge states or generative processes in another cycle. This requires that there be
   a. a test process that not only simply rejects or accepts a knowledge state as the appropriate solution, but also defines and measures its differences and similarities to other knowledge states.
   b. guidelines which indicate which generators and knowledge states to employ in the next cycle based upon the differences and similarities revealed in the test process.

3. Information systems grow out of particular complex problem solving methods. When the initial knowledge state (input) becomes fixed in its attributes (has the same form and meaning), the problem-solving method applied satisfactorily to that knowledge state before should again generate an acceptable final knowledge state; i.e., one acceptable to the test process (for example, generate a current inventory sheet which the problem solver can use to decide whether orders should be placed). If it appears the problem has some likelihood of recurrence, the problem solver will likely store or record that method of complex problem solving. If he does, an information system is born. Simply stated, an information system is a recorded (memorized, stored, or filed) particular method of complex problem solving.

4. Individual information systems become general information systems as they prove their value to a group of problem solvers. This usually happens because time shows the output of the system to be consistently useful to the test process. The use of the individual system by other people often leads to its modification and to its combination with other individual systems to create a more comprehensive information system.

B. Build an information center system which will facilitate all of the above:
   1. Simple and complex problem-solving
   2. Development and use of individual information systems.
   3. Recognition of individual systems having general application.
   4. Selection, combination, and refinement of individual systems into a general information system.

VI. What General Approach Should Be Followed?

A. Encourage, develop and incorporate user creativity; i.e., foster user prototyping.

1. Make the center's system easy for new users, but
2. Build in shortcuts for experienced users.
3. Don't make it difficult for the user to build new functions (generating processes) for his problem-solving.
4. Make it easy for the user to catalogue and use his systems.
5. Advertise and market the user's systems and functions to other users.

B. Convert to lower level languages (PL/I, Fortran, etc.) only those general systems that have stood the test of time, have matured.

C. Search for any new data source or function which could be useful to the user.

D. Know and appreciate the user's task and objective.

VII. How Should The Information Center's System Be Constructed?

A. Create independent primary functions (generators) [5,9].

1. Put no more than one function on each menu.
2. Create a command language syntax for each function [13,20].

B. Let the user organize the functions as desired [12].
   1. Establish a jump capability between menus.
   2. Develop a command language generator for the menus.
   3. Create an edit option for the commands generated.

C. Allow display of the output of any function at any time. Put simple versions of PROC PRINT and PROC CONTENTS in PF keys.

D. Make the use of the functions quite clear.
   1. Keep the approach consistent from menu to menu.
   2. Tie the common name of the menu to its command language and the function it performs; e.g., sort, select, sum.
   4. Make available longer message prompts if the user desires more information.
   5. Establish an on-screen tutorial for each function and provide examples in each tutorial, not only of the code, but also of the operation of that function.

E. Automate user manuals and arrange them by function name so that the user can quickly access information and additions and modifications can be made easily.

F. Make it easy to catalogue the function organizations (information systems) created by users.

G. Let the user dissect the functions and create new ones.
   1. Develop a code generator with edit facility [3,9], but provide for several levels,
      a. MACRO code as defined
      b. MACRO code as resolved for processing
   2. Provide manuals and consultants for explaining the code.
   3. Give option to see code as performed (SAS log).
   4. Establish good testing and documentation facilities.

H. Standardize, catalogue, and describe the new functions.

I. Give the user a means of tracking his information process in order to see what operations have been done to that point.

J. Organize and name libraries so that they describe general function performed; e.g., menus, commands, tutor, formats.

K. Organize and name library members so that additions can be made easily and relationships can be easily identified.
   1. Use an alpha-numeric name which corresponds to the index and sub-index position of the function; e.g., SAS11A0 would mean first menu of first function of first sub-index, SAS11B0 would mean second menu of first function of first sub-index.
   2. Use related alpha-numeric names for the command, its menus, and tutorials.

VIII. How Should the Information Center's System Be Organized?

A. Develop a primary index with branches to sub-indices.

B. Have each sub-index reference similar type functions

Examples might be:

1. Data entry
   a. File location and structure
   b. Field and format structure

2. Record, row, or observation operations
   a. Sorts
   b. Transforms
   c. Summaries
   d. Criteria selections

3. Column, field, or variable operations
   a. Deletions
   b. Mathematical manipulations
   c. Recoding (formatting)
   d. Substringing
   e. Concatenation
   f. Ordering

4. Data output
   a. File Structure
   b. File destination
   c. Titling

5. File manipulations
   a. merges
   b. concatenates
   c. updates

6. Catalogue of analysis systems, including graphical and statistical procedures.
   a. General
   b. User-specific

C. Build the menus so that aspects of the data can be transferred to them if the user requests it.

D. Allow for menu continuation when necessary

E. Provide for menu expansion through good naming conventions
IX. What Software Utilities Should It Have?

A. On-line directory of permanent general and individual data libraries, data sets, variable names, and format libraries [11].

B. On-line directory of temporary data sets and their variable contents.

C. Keyed access to data sets for faster on-line selects [27,28].

D. Macros which perform functions such as
   1. Multi-level, multi-variable summaries.
   2. Transforms of data sets to desired row/column structure.
   3. Anticipation of post-transform column structure for effective reporting [23].

X. What Personnel Will Make the Center's System a Success?

A. Most importantly, a manager who can market the system's promise, effectively describe its benefits, and fine tune the methods of costing its resources.

B. User documentation/training specialists who design and create the system's entry-level (menu-related) training, which should reside predominantly on the system's operating panels and supporting tutorials.

C. Consultant/technique analysts for middle-level education and assistance, which would involve both direct user support and the writing of manuals describing effective use of the system's functions and command language.

D. Information specialists for system development and very high-level education and assistance. They would instruct users in the direct use of the code generated by the command language, so that the users could develop new functions or systems. They would also understand the users' tasks and objectives sufficiently to decide
   1. When an individual system should become a general system
   2. When new or additional data sources could aid the user.

E. Application and system programmers to transfer to lower level languages the analysis systems which have become very structured (mature). They would also develop the procedural interfaces to the command language of the system, which, if the system was SAS-based, would involve writing PL/I or Fortran procedures.

F. Data coordinators to plan, develop, and maintain the data extraction methods which create the system's data bases.

XI. Why Use ISPF as the Primary Interface for the System?

A. ISPF itself is a good example of proper menu organization.

B. ISPF tables provide an easy way of passing information between SAS-based subsystems and COBOL, Fortran, and PL/I subsystems.

C. If the user is familiar with the ISPF language and utilities, education in the basic commands and approach of the system is easy.

XII. Why Use SAS/DMI® or SAS With IIF Instead of a Simple ISPF/SAS System

SAS/DMI or SAS with IIF facilitates an interactive menu application, which greatly assists problem solving by providing an easy method for reviewing and replacing the operations and data being used in a problem solving. An interactive menu ISPF/SAS system without SAS/DMI or SAS with IIF would require recurring initializations of the SAS system and thus would be both cumbersome and expensive to operate.

XIII. What's Different About A SAS/DMI® System?

An ISPF SAS/DMI system should not use the automatic implicit hierarchial menu structure of ISPF, as is usually the case in an ISPF/SAS batch system controlled by a TSO CLIST. That approach would require the same recurring initializations of the SAS system described above. Rather, SAS should be entered once and all dialog activities should be controlled by one or more SAS data steps [29, Appendix 4]. As a result, the system must use an explicit hierarchial chaining approach, such as used in the tutorial portion of ISPF [32, PP. 228-231]. One method of accomplishing this would be to load a global symbol table upon entry to the primary panel of the SAS/DMI system. This table would contain the panels to be displayed and actions to be performed when the user entered his panel selection on the command line. In addition, the table would contain the names of the parent index for that panel and the name of the previous panel, elements normally contained in the end and return commands of ISPF.

XIV. Where To Get More Details

See Bibliography.

Brackets [ ] indicate a source for further information on the subject, not necessarily the source of the conclusions or prescriptions presented regarding that subject. Furthermore, the articles referenced may certainly provide information on other subjects not indicated by the brackets.

XV. Acknowledgements

Particular thanks are to be given to Jim Watts and Dave Hundley for their willingness to describe in detail the purpose, approach, and functions of the system used by their information center in the dyes stuffs and chemicals division of CIBA-GEIGY Corporation, Greensboro, NC. They were an inspiration not
only to my understanding of what functions an information system should have, but also to my recognition of the attitudes and skills vital in an information center staff. Also, appreciation is extended to Mike Sutton of the Data Systems Division of General Dynamics for his enthusiastic readiness to discover and discuss the shortcomings of this paper.

SAS and SAS/AF are registered trademarks of SAS Institute Inc., Cary, NC, USA. SAS/DMII is a trademark of SAS Institute Inc. IIF is a product of Tangram Systems Corporation, Raleigh, NC, USA. ISPF is a product of International Business Machines Corporation.

For additional information, contact Jim DeFoor at 817-777-1590 or write P.O. Box 748, Mail Zone 1638, Department 18, General Dynamics, Fort Worth, TX, 76101.

Bibliography

Articles and Books on the Purpose and Structure of Information Systems.


Articles Giving General Guidance and Good Examples for the Design of an Information System


Articles Providing Specific Assistance On The Design of Various Tools or Elements of a Good SAS-Based Information System


Manuals


32. IBM Corporation, Interactive System Productivity Facility (ISPF), Dialog Management Services (DMS), Program Number 5668-960, for use with OS/VS2 MVS Release 3.8, VM/SP, VSE/AF Release 3.5*, Poughkeepsie, NY. 1983. 331 PP.