TITLE: "AUTOMATED SYSTEM DESIGN: FACT OR FICTION?"

by Tim Bryce
Managing Director
M. Bryce & Associates (MBA)
P.O. Box 1637
Palm Harbor, FL 34682-1637
United States
Tel: 727/786-4567
E-Mail: timb001@attglobal.net
WWW: http://www.phmainstreet.com/mba/
Since 1971: *"Software for the finest computer - the Mind"*

"If an information requirement is stated improperly to begin with, then everything else that follows will be incorrect."
- Bryce's Law

INTRODUCTION

Automated System Design is a fact, not fiction. I am not talking about some GUI-based program where you move boxes around on a screen. I am talking about a tool with the ability to interpret requirements and deduce a total system design. Some might call this an "expert system," I call it good old-fashioned common sense. Nor am I talking about a software engineering tool but, rather, a precursor to software engineering altogether performing the vital up-front design work that is sorely lacking in today's development world. Instead, I am describing a tool that can automatically design an enterprise-wide system, complete with data base. Not only does such a tool save time during design and documentation, it forces better software engineering specifications, and promotes data integration and a reduction of work effort. Free of design detail, Systems Engineers are thereby free to consider alternative designs and evaluate the use of packaged solutions (it provides a handy roadmap for selecting a commercial product).

I have described the differences between System Design and Software Design on many occasions; see the following "PRIDE" Special Subject Bulletins:

No, 23 - "*Using Logical Models as Templates*" - May 09, 2005
http://www.phmainstreet.com/mba/ss050509.pdf

No. 14 - "*What is a good program spec?*" - Mar 07, 2005
http://www.phmainstreet.com/mba/ss050307.pdf

No. 13 - "*Re-Inventing Business Process Design*" - Feb 28, 2005
http://www.phmainstreet.com/mba/ss050228.pdf

Automated System Design is not "smoke and mirrors" but, instead, a reality. Surprisingly, this technology has been available for the last 26 years with the advent of our Automated Design Facility (ADF) in 1979 (not to be confused with IBM's ADF). In 1987 we enhanced the product and renamed it ASE (Automated Systems Engineering). But since we put the "PRIDE" Methodologies for IRM in the public domain last November, I am now going to reveal how it is done. Pay close attention; I'm only going to explain it once. Here is how it works...

PREREQUISITES

In order to make Automated Systems Design (hereinafter "ASD") a reality, some ground rules have to be established. First, we have to make an admission that Systems Design is a teachable science based on some standard and fundamental principles derived from engineering; to wit:

A system is a product that can be engineered and manufactured like any other product. Following this principle, we invented the "Standard System Structure Concept" which defines the universally applicable architecture for all information systems:

A system consists of one or more sub-systems representing business processes existing within a specific time frame (Frequency, Offset, and Response Time). Each sub-system is implemented by a "work flow" of procedures; one or more administrative procedures (what the human-being performs) and one or no computer procedures. Administrative procedures consist of one or more operational steps (tasks), and; Computer Procedures consist of one or more programs.

This "Standard System Structure" is based on a product structure representing: the product, its assemblies, their sub-assemblies, and operations. Designing an information system, therefore, is essentially no different than building any other product. We design top-down, and implement and test bottom-up.

The only way any portion of the system structure interfaces with another portion is through shared data (representing the "parts" of the product). Data resources are organized into files, records and data elements. Inputs represent how data is collected, and outputs represent how data is retrieved. Pretty simple, right?

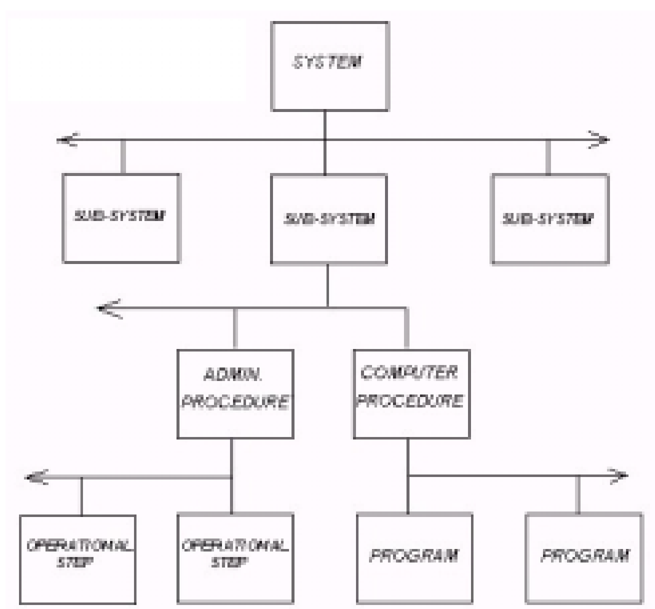
Layered documentation (flowcharts and text) is used to represent the design of each level in the system hierarchy which is analogous to "blueprinting" in engineering. For example:

- A System Flowchart shows sub-systems.
- A Sub-System Flowchart shows procedures.
- A Computer Procedure Flowchart shows programs.

Note: We do not flowchart Administrative Procedures as they are written according to a "playscript" technique.

(continued on page 2)

"PRIDE" STANDARD SYSTEM STRUCTURE CONCEPT



(continued from page 1)

These concepts have stood the test of time and have been proven in thousands of applications around the world, in every field of endeavor imaginable.

Next, systems design requires a methodical approach addressing each level in the system hierarchy. Such an approach should promote "design correctness" to assure a viable design. Instead of designing the system in its entirety all at once, step-wise refinement is needed to decompose each level of the system hierarchy into the next.

To decompose a system into sub-systems, "PRIDE" makes use of the concept of "Information Driven Design" whereby the intent is to work backwards from Information requirements, to the necessary primary data elements to be collected. The timing specifications for information, dictates when to collect, store and retrieve data which leads to the formation of sub-systems.

There are essentially three types of sub-systems: "Maintenance" sub-systems are used to collect data; "Display" sub-systems to reference data; and a combination of "Maintenance/Display" to read/write to the data base. Files associated with each sub-system are assigned in terms of how they are Created, Updated, or Referenced (C/U/R).

The sub-system represents logically "what" must occur and "when." Then, when we go to the next level of detail, where the sub-system is decomposed into its procedural work flow, the design effort moves forward, from start to end, defining "who" and "how" the data is to be physically processed. As procedures are defined, consideration is given to the three basic processing constructs: sequence (1, 2, 3, etc.), iteration (repeat until a condition is met), or choice (selection based on criteria). Whereas the sub-system represents a logical construct, the procedures represent physically how processing is to occur.

The decomposition of administrative procedures into operational steps and computer procedures into programs, follows a similar design philosophy (moving forward) except with more refinement of detail.

Finally, the last requirement is an IRM Repository (aka, "Data Dictionary") to catalog and cross-reference information resources, including the system and data resources mentioned above (as well as others). The purpose of the IRM Repository is to promote the sharing and re-use of information resources, thereby avoiding redundancies and promoting integration. The Repository is a software tool critical to the implementation of Automated Systems Design. Without it, ASD is not plausible.

For more information on the IRM Repository, please see.

"PRIDE" Special Subject Bulletin No. 10 ("*Managing Design Complexity*" - Feb 07, 2005)
http://www.phmainstreet.com/mba/ss050207.pdf

Establishing an IRM Repository
http://www.phmainstreet.com/mba/pride/spir.htm

If you do not accept the principles mentioned above, stop. Do not continue with this paper as you will not accept its conclusions. Let's not waste the time of either one of us.

PRELIMINARIES

Before we can pull the trigger on ASD, the Systems Engineer must perform some analysis and input certain specifications into the IRM Repository. This requires interviewing end-users and an in-depth understanding of the business problem. As such, the individual charged with this task must be skilled in analysis. Shoddy or superficial specifications will inevitably result in an inferior system design ("*you get what you pay for*"). From the analysis, the Systems Engineer must:

1. Define Information Requirements (IR) into the IRM Repository.

A textual description of each requirement is defined in terms of the Business Purpose, the Actions and/or Decisions to be supported, and the Benefits derived from the information. Further, the following specifications must be recorded:

- Information type: Policy (as used by executive management to establish direction), Control (as used by middle management to implement policy and oversee operations), and Operational (the day-to-day activities of the enterprise).
- The timing of the information: Frequency (how often the information is required), Offset (when to start), and Response Time (speed of delivery).
- The "receivers" of the information (end-users). In the Repository, we attach each IR to the assigned Organizational Entities (OE) representing the users.
- The required data elements to satisfy the information requirement, both new and existing (already defined in the IRM Repository). This includes both primary and generated elements (see Data Definitions in the next step).

For more information on defining information requirements (along with a handy worksheet), see "PRIDE" Special Subject Bulletin No. 4 (Dec 27, 2004)
http://www.phmainstreet.com/mba/ss041227.pdf

In "PRIDE", we also establish a Project Description (PD; to represent the system assignment) and attach the IR's to it.

2. Define Data Descriptions (DD) into the IRM Repository.

As mentioned, both primary and generated data elements are defined. Primary data refers to those elements inputted by a user; generated data refers to calculations or group items dependent on other data elements. Generated data must be defined backwards until all primary values are properly identified and defined.

At this time, it is not necessary to define the physical characteristics of each data element (e.g., length, picture, program label, etc.) but, rather, just its logical definition, including:

- Business definition - a textual description expressed in terms the end-user will understand (expressed in a dictionary-like format).
- Purpose - describing how the data element is used; Indicative (used for identification purposes), Descriptive (describes the characteristics of an object), or Quantitative (numeric values that are either calculated or are calculable).
- Source - for primary data, the source is represented by the user area ultimately responsible for its assignment. For generated data, the data elements required to produce it (e.g., Data-1 + Data-2 = Data-3); "Net Pay," "Balance Amount" and "Percent Complete" are some examples of calculated data.

For more information on defining data elements, see: "PRIDE" Special Subject Bulletin No. 34 ("*The Benefits of a Data Taxonomy*" - July 25, 2005)
http://www.phmainstreet.com/mba/ss050725.pdf

(continued on page 3)

(continued from page 2)

Finally, attach the DD's to their assigned Information Requirements in the IRM Repository.

3. Define Output Descriptions (OD) into the IRM Repository.

Outputs represent the vehicles by which information is delivered to the user. They can also take many forms: printed reports, screens, audio response, etc. Further, there is not necessarily a one-to-one relationship between information requirements; one information requirement may require multiple outputs, and one output may support multiple information requirements. Nonetheless, the Systems Engineer determines the Outputs necessary to satisfy each information requirement and documents it as an Output Description (OD) in the IRM Repository. At this time, it is not necessary to physically layout the output, but, rather, to describe it as:

- Textual description stating the Business Purpose of each output.
- Timing characteristics - Frequency, Offset, Response Time.
- The data element(s) used as the key to the output; e.g., Product Number, Employee Number, etc.
- Attach the DD's to be used in each output.

Attach the OD's to the IR's they support in the IRM Repository.

As an aside, if by chance you know how you want the output to be implemented, such as in an office-automation procedure (e.g., fax transfer), or manually prepared, there are overriding switches in "PRIDE" that can be tagged to the OD which will influence the design. Otherwise, let the software figure it out for you.

4. Define logical Record Descriptions (RD) into the IRM Repository.

For all of the primary data elements defined thus far, organize them into logical records (not physical storage records) based on their "basic grouping" (BG) representing logical keys. Note: logical records do not include generated data elements as they are calculated within the system based on primary elements.

For more information on "Basic Grouping," see:
http://www.phmainstreet.com/mba/pride/dbmeth.htm#bgconcept

All logical records, should be assigned a "source" indicating the end-user area responsible for inputting the data.

Attach the RD's to the OD's they support in the IRM Repository.

5. Define a System component in the IRM Repository (representing the highest level in the System hierarchy).

Attach the OD's to the System component. Note: If you want the ASD to modify an existing system, note the outputs to be considered (and those to be ignored). Otherwise, the ASD will re-design the entire system.

Attach the System component to the IR's and PD.

Finally, the information resources in the IRM Repository represent the specifications needed to perform Automated Systems Design. This also means the Repository is the data base to be referenced and updated as the ASD tool performs its task.

PULLING THE TRIGGER

The following describes the processing logic used in Automated Systems Design:

I. Check input parameters (as specified by the Systems Engineer):

A. Check the job key which defines the associated PD and System component affected:

If a System is entered, only jobs involving a System Design will be allowed.

If a Sub-System is entered, only jobs involving a Sub-System Design will be allowed (and not System Design).

If a Computer Procedure is entered, only the Computer Procedure design will be allowed (and no others).

B. If modifying an existing system, identify the OD's to be considered.

C. In "PRIDE", all information resources are identified by name and control number. To this end, when ASD creates new components, give it a set of starting numbers to build components.

D. Select Design Level - choices include:

- Perform System Design (designs sub-systems), Sub-System Design (designs procedures), and Computer Procedure Design (designs programs) (Default parameter).

- Perform System Design only.

- Perform System Design and Sub-System Design only.

- Perform Sub-System Design and Computer Procedure Design only for a single sub-system.

- Perform Sub-System Design only for a single sub-system.

- Perform Computer Procedure Design only for a single computer procedure.

E. Select the type of processing desired:

- Design both Interactive and Batch processes - this is the default operating parameter. ASD will design suitable sub-systems as it deems necessary.

- Design a totally "Batch" type of system - directs ASD to design only batch sub-systems.

- Design a totally "Interactive" type system - directs ASD to design only interactive sub-systems.

F. The Systems Engineer may select the following design options:

- MERGING MAINTENANCE SUB-SYSTEMS - This option is particularly useful to minimize the number of file maintenance sub-systems. The Systems Engineer should also consider the use of the "Combine records into inputs based in Basic Grouping only" when exercising this option.

Note: Use of this option voids the use of the "Combine Output Processing with Input Processing"

- MERGE DISPLAY SUB-SYSTEMS BASED ON TIMING ONLY - ASD will normally group OD's into the same sub-system

(continued on page 4)

(continued from page 3)

based on compatible timing (Frequency/Offset/Response Time) and "Receivers" (the receiver of the output). This option voids the use of "Receivers" and will minimize the number of "Display" type sub-systems.

Note: Use of this option voids the use of the "Combine Output Processing with Input Processing"

- COMBINE OUTPUT PROCESSING WITH INPUT PROCESSING - ASD will design separate Display and Maintenance sub-systems by default. These can be merged manually later on using other edit facilities. However, this option provides the means to combine outputs with the inputs that collect the data used to produce the information. ASD will combine the processes based on identical timing (F/O/R).

Note: If either the "Merge Maintenance Sub-Systems" option or "Merge Display Sub-Systems based on timing only" option is activated, this option will be void.

- COMBINE RECORDS INTO INPUTS BASED ON BASIC GROUPING ONLY - ASD will normally combine records into a single input based on common Basic Grouping and record "source" (the user who must input the data). However, this option combines records into inputs based on the primary Basic Grouping (first BG DD) but not user "source". This minimizes the number of inputs used during file maintenance by placing multiple RD's on to a single ID. This is particularly useful when exercising the "Merging Maintenance Sub-Systems" option.

- BASIC SUB-SYSTEM DESIGN ONLY - normally, ASD will build sub-systems with as many procedures (both computer and administrative) as it deems necessary. However, this option minimizes the number of procedures necessary to execute a sub-system.

II. Check the specifications of information resources as recorded in the IRM Repository (see "Preliminaries" section above). Errors in component definition suspend execution. A "Pre-Scan" report is produced for the Systems Engineer to correct errors in resource definitions.

A. Select continued path based on specified operating parameter.

1. Go to Section III for a complete System Design.

2. Go to Section VI for just a Sub-System Design.

3. Go to Section VII for just a Computer Procedure

III. Read OD specifications.

A. Read RD's attached to OD's.

1. Merge RD's with identical BG, Source and DD lists. (Use the RD that was found first).

a. Make substitutions to OD/RD relationships.

2. For OD's, merge RD's with identical BG, Source, regardless of DD list, only if the frequency and offset of the OD's are identical. (Use the RD with the most DD's or the first found in case of a tie).

a. Make substitutions to OD/RD relationships.

B. Create new ID's for each RD.

1. Use the starting ID number as stated for the job.

a. Perform IRM Repository component number duplication check. (This is performed for all newly created components).

2. Attach RD's to ID. At this point, no more than one RD may be attached to an ID.

3. Define the timing of the ID identical to the OD where the RD originated from.

4. Assign the Primary Basic Grouping of the ID; this comes from the RD.

C. Merge RD's into a single new RD, wherever possible, based exclusively on Basic Grouping (BG).

1. Use the starting RD number as stated for the job.

a. Perform IRM Repository component number duplication check.

2. Add the RD to the ID's and OD's where the former RD's originated from.

a. For each RD added to the ID, add '++ASD' to the ID/RD relationship.

D. Group RD's into new "application logical" File Description's (FD) based on the same Primary Basic Grouping of the RD's.

1. Assign the Primary Basic Grouping to the FD.

E. Prepare tentative "display" type sub-systems for the OD's.

1. Read "Combine Display Sub-System Option."

a. If "N" (No), group OD's by compatible timings (F/O/R) and "Receivers."

b. If "Y" (Yes), group OD's by compatible timings (F/O/R) only (not by "Receivers").

2. Special ASD Design Keyword considerations.

a. If keyword "MANUAL" found, separate from other OD's and group based on compatible timings (F/O/R) and "Receivers",

b. If "OFFICE-AUTO" ("OA") found, separate from other OD's and group based on compatible timings (F/O/R) and "Receivers".

3. Attach supporting application logical FD's to the pertinent sub-systems. This is determined by the records in the FD's that are used to support the OD's.

F. Prepare tentative "maintenance" type sub-systems based on unique time frame of the ID's.

1. Attach ID's and supporting application logical FD's. At this point, no more than one input and file can be placed in a single sub-system.

(continued on page 5)

(continued from page 4)

G. Print "ASD New Logical File Design" for the Systems Engineer's review (explains the work thus far).

IV. For each new FD, search the IRM Repository for existing application logical FD's matching the Primary Basic Grouping.

A. If no match, go to step V.

B. Match each RD in the new FD to the RD's found on the existing FD's. Comparison is based exclusively on Basic Grouping.

1. If no match, proceed to step V.

C. Check DD content of each RD match. Select the RD with the highest percentage match.

1. If 1% to 100% match, attach the existing FD to the new tentative maintenance sub-system with a ++ASD tag in the Sub-System/FD relationship.

2. If no match, proceed to step V.

D. Search the IRM Repository for sub-systems that update each existing FD identified in each tentative "maintenance" sub-system.

1. List the ID's used to update the file along with their timing.

2. Match the RD's on the ID's with the RD on the tentative ID (the RD without the ++ASD tag).

a. RD's must first have same BG and "source."

1. If no match, tentative "maintenance" sub-system remains.

b. Compare DD content of each RD.

1. If no match, tentative "maintenance" sub-system remains.

2. If match, delete the DD from the RD on the new ID as they are found.

- If all DD's are deleted, delete the RD from the new ID.

- If the RD is deleted from the ID, delete the new ID.

- If the ID is deleted, delete the tentative "maintenance" sub-system.

- Delete the RD entirely only if it doesn't reside on an FD.

2. Any remaining DD's are left in the RD.

d. Check timing of ID's.

1. If compatible (same or less)

- Attach remaining RD to existing ID.

- Delete the tentative ID.

- Print message that the existing ID has been modified.

- Delete the tentative "maintenance" sub-system.

2. If incompatible (greater than)

- Attach RD to existing ID.

- Delete the tentative ID.

- Delete the tentative "maintenance" sub-system.

- Change timing of sub-system that the existing ID is used in (must have Systems Management approval).

G. List all of the remaining sub-systems in the system along with the existing sub-systems modified.

H. Print "ASD File Analysis" for the System Engineer's review (explains the design thus far).

V. Prepare final system design.

A. Review ASD Operating Options.

B. Consider option to merge OD with ID in same sub-system.

1. If all necessary ID's and FD's reside in the same sub-system, merge the OD into sub-system.

2. If not, do not merge.

C. For those "display" type sub-systems without an ID, create and add a request type ID.

D. For those "maintenance" type sub-systems without an OD, create and add a "data validation" type OD.

E. Print the final sub-systems showing timing ID's, FD's and OD's on the "ASD System Design" for the Systems Engineer's review.

F. Select next step.

1. Proceed to Section VI for ASD Jobs 3 & 4.

2. Proceed to Section VIII for ASD Job 2 and Flowcharting option is 1 or 2.

3. End processing for ASD Job 2 and Flowcharting option is "N" (No).

VI. Perform Sub-System Design.

A. Read pertinent ASD Operating Parameters.

1. Note whether a "basic" or "advanced" sub-system design is required.

2. Note whether selected OD's or ID's have any special keyword specifications to force a manual design, or office automation design. (ASD will try to design a computer assisted system unless otherwise noted).

(continued on page 6)

(continued from page 5)

B. Interpret OD/ID specifications.

1. If response time is expressed in seconds, sub-system will be interpreted as interactive.

2. If response time is expressed in anything other than seconds, sub-system will be interpreted as batch.

C. Select desired type of sub-system: Basic or advanced.

1. For Basic, go to step VI - D.

2. For Advanced, go to step VI - E.

D. For Basic Sub-System Design, determine operation to files.

1. If primary file is being created, updated and referenced (maintenance) interactively, use the following procedures:

   a. Terminal Session (AP) - number 03

   b. Computer File Read/Write (CP) - number 05

2. If primary file is only being referenced (displayed) interactively, use the following procedures:

   a. Terminal Session (AP) - number 03

   b. Computer File Read (CP) - number 05

3. If primary file is being created, updated and referenced (maintenance) in batch, use the following procedures:

   a. Input Preparation (DC) - number 03

   b. Computer File Read/Write (CP) - number 05

   c. Output Review (AP) - number 07

4. If primary file is only being referenced (displayed) in batch, use the following procedures:

   a. Input Preparation (DC) - number 03

   b. Computer File Read (CP) - number 05

   c. Output Review (AP) - number 07

5. If primary file is being referenced only (displayed) through manual means only ("MANUAL" override on OD/ID), use the following procedures:

   a. Manual File Look-Up (AP) - number 03

   b. Output Review (AP) - number 05

6. If primary file is being created, updated and referenced (maintenance) through manual means only ("MANUAL" override on OD/ID), use the following procedures:

   a. Input Preparation (AP) - number 03

   b. Manual File Update (AP) - number 05

7. If "OFFICE-AUTO" ("OA") override on OD/ID, use the following procedures:

   a. Office Automation (OA) - number 03

E. For Advanced Sub-System Design, determine operation to files.

1. If primary file is being created, updated and referenced (maintenance) interactively, use the following procedures:

   a. Terminal Session (AP) - number 03

   b. Computer File Read/Write (CP) - number 05

2. If primary file is only being referenced (displayed) interactively, use the following procedures:

   a. Terminal Session (AP) - number 03

   b. Computer File Read (CP) - number 05

3. If primary file is being created, updated and referenced (maintenance) in batch, use the following procedures:

   a. Input Preparation (AP) - number 03

   b. Data Conversion (DC) - number 05

   c. Computer File Read/Write (CP) - number 07

   d. Output Distribution (AP) - number 09

4. If primary file is only being referenced (displayed) in batch, use the following procedures:

   a. Input Preparation (AP) - number 03

   b. Data Conversion (DC) - number 05

   c. Computer File Read (CP) - number 07

   d. Output Distribution (AP) - number 09

   e. User Action/Decision (AP) - number 11

5. If primary file is being referenced only (displayed) through manual means only ("MANUAL" override on OD/ID), use the following procedures:

   a. Manual File Look-Up (AP) - number 03

   b. User Action/Decision (AP) - number 05

6. If primary file is being created, updated and referenced (maintenance) through manual means only ("MANUAL" override on OD/ID), use the following procedures:

   a. User Action/Decision (AP) - number 03

   b. Input Preparation (AP) - number 05

   c. Manual File Update (AP) - number 07

(continued on page 7)

(continued from page 6)

8. If "OFFICE-AUTO" ("OA") override on OD/ID, use the following procedures:

   a. Input Preparation (AP) - number 03

   b. Office Automation (OA) - number 05

   c. Output Review (AP) - number 07

C. Create necessary Working Files. All are considered application physical files and their Logical/Physical Indicators should be marked "Physical."

1. Create an Input Transaction File (computer) for all batch processing with a computer procedure. Attach this file between the Input Preparation and Computer Procedure in a "basic" design, and; attach the file between a Data Conversion procedure and a Computer Procedure in "advanced" design.

2. Create a Data Conversion Hold File (manual) when a Data Conversion procedure is present. Attach this file to Data Conversion and Output Review procedures.

3. Create a Document Storage File to store all paper reports.

4. Create a Computer Output File to be generated from a computer procedure so it coincides with the corresponding OD containing "FILE" as a keyword.

D. Print "ASD Sub-System Design" for the System Engineer's review (explains the design thus far).

E. Select next step.

1. Proceed to Section VII if so indicated.

2. Proceed to Section VIII if so indicated.

3. End processing.

VII. Perform Computer Procedure Design:

A. Review Computer Procedure specifications.

1. Check response time of Procedure.

   a. If timing is expressed in seconds, will assume interactive processing.

   b. If timing is expressed in anything other than seconds, will assume batch processing.

2. Check operation to primary file (Create/Update/Reference).

   a. If interactive and file is created, updated and referenced, create one program entitled File Update.

   b. If interactive and file is referenced only, create one program entitled Generate Output(s).

   c. If batch and file is created, updated and referenced, create one program for Data Validation, and one program for File Update.

   - Create a Program Work File to communicate between the two programs. This is a working file that is application physical.

   d. If batch and file is referenced only, create one program to Generate Output(s).

C. Print "ASD Computer Procedure Design" for the Systems Engineer's review (explains the design thus far).

D. Select next step.

1. Proceed to Section VIII if so indicated.

2. End processing.

VIII. Print Flowcharts for the Systems Engineer's review:

A. System Flowchart (showing sub-systems).

B. Sub-System Flowchart (showing procedures).

C. Computer Procedure Flowchart (showing programs).

For more information on "PRIDE" flowcharting, see:

"PRIDE" Special Subject Bulletin No. 13 - ("*Re-Inventing Business Process Design*" - Feb 28, 2005)
http://www.phmainstreet.com/mba/ss050228.pdf

"PRIDE" Flowcharting Symbols
http://www.phmainstreet.com/mba/pride/isspfs.htm

IX. Miscellaneous Considerations

A. Naming Conventions

1. For generated ID's used for maintenance:

   a. Take the 8 character source field of the RD contained on the ID and add "Input."

   Example:
   SALES INPUT

2. For generated ID's used to select a display:

   a. Use the word "Query" and the first 24 characters of the OD it supports. If more than one OD is used, use the first OD encountered.

   Example:
   QUERY CUSTOMER ORDER

3. For new application logical FD's (primary):

   a. Use the first word or 17 characters of the first word primary basic grouping DD followed by "Logical File."

   Example:
   Primary Key = Customer Number
   FD name = CUSTOMER LOGICAL FILE

(continued on page 8)

(continued from page 7)

4. For new RD's resulting from a merger:

   a. Use the first word or 15 characters of the first word primary basic grouping DD followed by "Logical File."

   Example:
   Primary Key = Customer Number

   FD name = CUSTOMER LOGICAL RECORD

5. For new OD's that support a maintenance ID; simply use the name DATA VALIDATION MESSAGE

B. Other Considerations

1. Use of the "IGNORE" keyword applied on any component will prohibit components from being considered in design. This is particularly useful for FD's.

NOTE: This program generates entries in a Transaction Hold File (Shadow File) thereby prohibiting direct updates to the IRM Repository before committing to a specific design (thereby saving time undoing design changes). Upon re-execution of ASD, a different Hold File should be accessed to allow multiple versions of the system design. This is only needed if analyzing multiple versions of design.

X. EXIT.

IN A NUTSHELL

I have gone into great detail to explain the processing logic of ASD. In summary, ASD first creates a rudimentary design then slashes away at it based on the operating parameters and renders a "correct" design for review by the Systems Engineer. In the process, it evaluates the use of existing information resources as recorded in the IRM Repository and, wherever possible, re-uses existing components, thereby avoiding the "reinventing the wheel" phenomenon (data redundancy) and encouraging systems integration. When completed, the Systems Engineer can either make minor modifications to the design using the IRM Repository editing routines or changing some ASD operating parameters and creating an alternative design.

The files created by ASD are considered "application logical" as defined by "PRIDE", suitable for any physical implementation (DBMS of otherwise). It also creates "application physical" files where appropriate to pass data between programs or procedures (e.g., input transaction files, output data files).

ASD also automatically populates the IRM Repository with component definitions based on canned names and descriptions. These should be refined by the Systems Engineer. As an aside, these names and descriptions are maintained in a separate flat file suitable for translation to accommodate foreign languages.

Although ASD is primarily intended for new designs, it can also be used to modify existing systems or document existing systems. As to the latter, a Systems Engineer collects specifications about an existing system (primarily OD, RD, and DD components), then runs ASD to document the existing system. This is a slick way of documenting your current systems.

There are several operating parameters which I have alluded to in the processing logic to force ASD to create manually implemented sub-systems (no computer procedures), office-automation sub-systems, or to force it to IGNORE certain resources in the IRM Repository (such as certain files).

What this means is the ASD has several operating parameters for the Systems Engineer to influence the ASD and express his creativity accordingly. Further, the ASD populates the IRM Repository with hundreds or thousands of transactions that would normally take days to perform otherwise. In the end, the system is fully documented and cataloged in the IRM Repository along with all of the company's other information resources. ASD, therefore, is a powerful documentation aid.

CONCLUSION

Performing a thorough systems design is a tedious task. ASD is a tool that takes the drudgery out of systems design and provides the Systems Engineer with a means to consider multiple designs. Now, instead of one design (that lacks confidence), you can produce multiple designs, all of which are plausible. After ASD produces a design, the Systems Engineer must consider the feasibility of its implementation in his company. If it isn't, the Systems Engineer needs to either modify the design or re-execute the ASD tool.

ASD performs the grunt work of systems design, but there are still many things to do in order to complete the design: Outputs and inputs still have to be firmed up as do the data resources, along with their physical implementation, supplemental text has to be written, administrative procedures have to be detailed, etc. But ASD goes a long way in terms of producing a design the Systems Engineer can have confidence in.

This approach also ties business process design with software engineering. There are those in the I.T. community who suggest these are distinctly separate activities; that software should be devised before the system (I call this the "solution looking for a problem" phenomenon). As for me, I'm a top-down man who does not want to waste the time of his programmers. By using a layered approach, the systems design drives the software design, not the other way around. This results in better specifications for Software Engineers to build programs to satisfy business needs.

The purpose of this paper is to not necessarily sell anything, but to make a point. In order to make Automated Systems Design a reality, you must have your act together. First, there has to be an admission by your company that systems design is a teachable science, complete with standard concepts and philosophies based on sound principles for design. Otherwise, it is not feasible.

For those naysayers who think ASD is not possible, and that system design is more of a creative process relying more on human intuition than anything else, I say simply, "*Bullshit*." Automated Systems Design is a reality; but it all depends on our perspective: we either view it as a science or an exotic art form. I choose the former.

Regardless of the designs produced by ASD, it all ultimately depends on how well information requirements are specified. If the requirements are wrong, then everything that follows will be wrong. There is simply no magic.

END

(continued on page 9)

"PRIDE" Special Subject Bulletins can be found at:

http://www.phmainstreet.com/mba/mbass.htm

They are also available through the "PRIDE Methodologies for IRM Discussion Group" at:

http://groups.yahoo.com/group/mbapride/

You are welcome to join this group if you are so inclined.

"PRIDE" is the registered trademark of M. Bryce & Associates (MBA) and can be found on the Internet at:

http://www.phmainstreet.com/mba/pride/pride.htm