

TITLE: "INFORMATION SYSTEMS THEORY 101"

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@phmainstreet.com

WWW: <http://www.phmainstreet.com/mba/>

Since 1971: *"Software for the finest computer - the Mind"*

*"The first on-line, real-time, interactive, data base system was double-entry bookkeeping which was developed by the merchants of Venice in 1200 A.D."
- Bryce's Law*

INTRODUCTION

Systems work is not as hard as you might think. However, we have a tendency in this business to complicate things by changing the vocabulary of systems work and introducing convoluted concepts and techniques, all of which makes it difficult to produce systems in a consistent manner. Consequently, there is a tendency to reinvent the wheel with each systems development project. I believe I owe it to my predecessors and the industry overall to describe basic systems theory, so that people can find the common ground needed to communicate and work. Fortunately, there are only four easy, yet important, concepts to grasp which I will try to define as succinctly as possible.

1. THERE ARE THREE INHERENT PROPERTIES TO ANY SYSTEM

Regardless of the type of system, be it an irrigation system, a communications relay system, an information system, or whatever, all systems have three basic properties:

A. A system has a purpose - such as to distribute water to plant life, bouncing a communications signal around the country to consumers, or producing information for people to use in conducting business.

B. A system is a grouping of two or more components which are held together through some common and cohesive bond. The bond may be water as in the irrigation system, a microwave signal as used in communications, or, as we will see, data in an information system.

C. A system operates routinely and, as such, it is predictable in terms of how it works and what it will produce.

All systems embrace these simple properties. Without any one of them, it is, by definition, not a system.

For our purposes, the remainder of this paper will focus on "information systems" as this is what we are normally trying to produce for business. In other words, the development of an orderly arrangement or grouping of components dedicated to producing information to support the actions and decisions of a particular business. Information Systems are used to pay employees, manage finances, manufacture products, monitor and control production, forecast trends, process customer orders, etc.

If the intent of the system is to produce information, we should have a good understanding of what it is...

2. INFORMATION = DATA + PROCESSING

Information is not synonymous with data. Data is the raw material needed to produce information. Data by itself is meaningless. It is simply a single element used to identify, describe or quantify an object used in a business, such as a product, an order, an employee, a purchase, a shipment, etc. A data element can also be generated based on a formula as used in a calculation; for example:

$$\text{Net-Pay} = \text{Gross-Pay} - \text{FICA} - \text{Insurance} - \text{City-Tax} - \text{Union-Dues} - (\text{etc.})$$

Only when data is presented in a specific arrangement for use by the human being does it become information. If the human being cannot act on it or base a decision from it, it is nothing more than raw data. This implies data is stored, and information is produced. It is also dependent on the wants and needs of the human being (the consumer of information). Information, therefore, can be defined as *"the intelligence or insight gained from the processing and/or analysis of data."*

The other variable in our formula is "processing" which specifies how data is to be collected, as well as its retrieval in order to produce information. This is ultimately driven by when the human being needs to make certain actions and decisions. Information is not always needed "upon request" (aka "on demand"); sometimes it is needed once daily, weekly, monthly, quarterly, annually, etc. These timing nuances will ultimately dictate how data is collected, stored, and retrieved. To illustrate, assume

(continued on page 2)

(continued from page 1)

we collect data once a week. No matter how many times during the week we make a query of the data base, the data will only be valid as of the last weekly update. In other words, we will see the same results every day for one week. However, if we were to collect the data more frequently, such as periodically throughout the day, our query will produce different results throughout the week.

Our formula of "I = D + P" makes an important point: if the data is changed, yet the processing remains the same, the information will change. Conversely, if the data remains the same, yet the processing changes, the information will also change. This leads to a compelling argument to manage data and processing as separate by equal resources which can be manipulated and reused to produce information as needed.

3. SYSTEMS ARE LOGICAL IN NATURE AND CAN BE PHYSICALLY IMPLEMENTED MANY DIFFERENT WAYS

An information system is a collection of processes (aka, "sub-systems") to either collect and store data, to retrieve data and produce information, or a combination of both. The cohesive bond between these components is the data which should be shared and reused throughout the system (as well as other systems). You will observe we have not yet discussed the most suitable way to physically implement the processes, such as through the use of manual processes, computer programs, or other office technology. In other words, at this stage, the sub-systems of the system simply define logically WHAT data must be processed, WHEN it must be processed, and who will consume the information (aka "end-users"), but it most definitely does not specify HOW the sub-system is to be implemented.

Following this, developers determine a suitable approach for physically implementing each sub-system. This decision should ultimately be based on practicality and cost effectiveness. Sub-systems can be implemented using manual procedures, computer procedures (software), office automation procedures, or combinations of all three. Depending on the complexity of the sub-system, several procedures may be involved. Regardless of the procedures selected, developers must establish the precedent relationships in the execution of the procedures, either sequentially, iteratively, or choice (thereby allowing divergent paths). By defining the procedures in this manner, from start to end, the developers are defining the "work flow" of the sub-system, which specifies HOW the data will be physically processed (including how it is to be created, updated, or referenced).

Defining information systems logically is beneficial for two reasons:

- It provides for the consideration of alternative physical implementations. How one developer designs it may very well be different than the next developer. It also provides the means to effectively determine how a purchased software package may satisfy the needs. Again, the decision to select a specific implementation should be based on practicality and cost justification.
- It provides independence from physical equipment, thereby simplifying the migration to a new computer platform. It also opens the door for system portability, for example; our consulting firm helped a large Fortune 500 conglomerate design a single logical payroll system which was implemented on at least three different computer platforms as used by their various operating units; although they physically worked differently, it was all the same basic system producing the same information.

These logical and physical considerations leads to our final concept...

4. A SYSTEM IS A PRODUCT THAT CAN BE ENGINEERED AND MANUFACTURED LIKE ANY OTHER PRODUCT.

An information system can be depicted as a four level hierarchy (aka, "standard system structure"):

- LEVEL 1 - System
- LEVEL 2 - Sub-systems (aka "business processes")
- 2 or more
- LEVEL 3 - Procedures (manual, computer, office automation) - 1 or more for each sub-system
- LEVEL 4 - Programs (for computer procedures), and Steps (for all others) - 1 or more for each procedure

Click for diagram:

<http://www.phmainstreet.com/mba/pride/issss.jpg>

Each level represents a different level of abstraction of the system, from general to specific (aka, "Stepwise Refinement" as found in blueprinting). This means design is a top-down effort. As designers move down the hierarchy, they finalize design decisions. So much so, by the time they finish designing Level 4 for a computer procedure, they should be ready to write program source code based on thorough specifications, thereby taking the guesswork out of programming.

(continued on page 3)

(continued from page 2)

The hierarchical structure of an information system is essentially no different than any other common product; to illustrate:

- LEVEL 1 - Product
- LEVEL 2 - Assembly - 2 or more
- LEVEL 3 - Sub-assembly - 1 or more for each assembly
- LEVEL 4 - Operation - 1 or more for each sub-assembly

Again, the product is designed top-down and assembled bottom-up (as found in assembly lines). This process is commonly referred to as design by "explosion" (top-down), and implementation by "implosion" (bottom-up). An information system is no different in that it is designed top-down, and tested and installed bottom-up. In engineering terms, this concept of a system/product is commonly referred to as a "four level bill of materials" where the various components of the system/product are defined and related to each other in various levels of abstraction (from general to specific).

This approach also suggests parallel development. After the system has been designed into sub-systems, separate teams of developers can independently design the sub-systems into procedures, programs, and steps. This is made possible by the fact that all of the data requirements were identified as the system was logically subdivided into sub-systems. Data is the cohesive bond that holds the system together. From an engineering/manufacturing perspective it is the "parts" used in the "product." As such, management of the data should be relegated to a separate group of people to control in the same manner as a "materials management" function (inventory) in a manufacturing company. This is commonly referred to as "data resource management."

This process allows parallel development, which is a more effective use of human resources on project work as opposed to the bottleneck of a sequential development process. Whole sections of the system (sub-systems) can be tested and delivered before others, and, because data is being managed separately, we have the assurance it will all fit together cohesively in the end.

The standard system structure is also useful from a Project Management perspective. First, it is used to determine the Work Breakdown Structure (WBS) for a project complete with precedent relationships. The project network is then used to estimate and schedule the project in part and in full. For example, each sub-system can be separately priced and scheduled, thereby giving the project sponsors the ability to pick and chose

which parts of the system they want early in the project.

Click for diagram:
<http://www.phmainstreet.com/mba/pride/network.jpg>

The standard system structure also simplifies implementing modification/improvements to the system. Instead of redesigning and reconstructing whole systems, sections of the system hierarchy can be identified and redesigned, thereby saving considerable time and money.

This analogy between a system and a product is highly credible and truly remarkable. Here we can take a time-proven concept derived from engineering and manufacturing and apply it to the design and development of something much less tangible, namely, information systems.

CONCLUSION

Well, that's it, the four cardinal concepts of Information Systems theory. I have deliberately tried to keep this dissertation concise and to the point. I have also avoided the introduction of any cryptic vocabulary, thereby demonstrating that systems theory can be easily explained and taught so that anyone can understand and implement it.

Systems theory need not be any more complicated than it truly is.

If you would like to discuss this with me in more depth, please do not hesitate to send me an e-mail at timb001@phmainstreet.com

Keep the faith.

REFERENCES

- 1 - "PRIDE" Methodologies for IRM
<http://www.phmainstreet.com/mba/pride/>
- 2 - "PRIDE" Special Subject Bulletins from MBA
<http://www.phmainstreet.com/mba/mbass.htm>
- 3 - "Understanding Information" by Tim Bryce
"PRIDE" Special Subjects Bulletin #78, June 5, 2006
<http://www.phmainstreet.com/mba/ss060605.pdf>
- 4 - "Defining Information Requirements" by Tim Bryce
"PRIDE" Special Subjects Bulletin #4, December 27, 2004
<http://www.phmainstreet.com/mba/ss041227.pdf>

(continued on page 4)

(continued from page 3)

5 - "Standard System Structure" by Tim Bryce -
January 18, 2008
<http://www.articlesbase.com/software-articles/standard-system-structure-308845.html>

6 - "Stepwise Refinement" by Tim Bryce - May 29, 2008
<http://www.articlesbase.com/programming-articles/stepwise-refinement-431223.html>

7 - "Understanding the Specifications Puzzle" by Tim Bryce - February 21, 2006
<http://it.toolbox.com/blogs/irm-blog/understanding-the-specifications-puzzle-22621>

(For Milt and Les).

END

About the Author

Tim Bryce is the Managing Director of M. Bryce & Associates (MBA) of Palm Harbor, Florida and has 30 years of experience in the field of Information Resource Management (IRM). He is available for lecturing, training and consulting on an international basis.

His writings can be found at:

<http://www.phmainstreet.com/timbryce.htm>

The "Management Visions" Internet audio broadcast is available at:

<http://www.phmainstreet.com/mba/mv.htm>

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

<http://www.phmainstreet.com/mba/pride/>

Copyright © 2009 by Tim Bryce. All rights reserved.