TITLE: “METHODS OF PROCESSING”

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"*

"A program without any form of transaction serves no useful business purpose."
- Bryce's Law

INTRODUCTION

As you travel around the IT industry, be it in a discussion group or attending a software conference, the general consensus is that all processing must be "interactive" or "real-time"; and "batch" processing is considered passe. Is it possible developers are overlooking other more important and inherent properties of processing? Frankly, I believe we are more enamored with the slick terminology of the industry as opposed to having a clear understanding of how processing works. Let's consider how data is truly processed.

TRANSACTIONS

In "PRIDE" Special Subject Bulletin No. 35 (*"Automated System Design: Fact or Fiction?"* - Aug 01, 2005) I went to great lengths to explain how a system design is performed. In the tutorial, I described how a system consists of a series of sub-systems (aka "business processes") implemented by administrative and computer processes. I went on to say,

"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)."

This scenario implies the use of some form of transaction, whether to request an output, or to input data to a file. From this perspective, a transaction is an exchange event from one object to another. A transaction always has some form of action associated with it. For output reporting, "request," "display," "print," "extract," "search," etc. are common transactions. For input, "new," "add," "change," "delete," "update," "charge," "credit," "debit," "deposit," etc. are typical transactions.

Over the years, transaction processing has come to mean "batch" processing as associated with 80 column records (anyone remember punch cards?), and typically relate to payroll, inventory or order processing. Not true. ALL processing involves the use of transactions, either one at a time (as in "interactive") or in groups ("batch").

What then becomes important is the volume of transactions and the speed they must be processed (as specified by timing). This will determine the physical constraints of the equipment to be used. "Batch" processing has the advantage of processing high volumes of transactions within a relatively short period of time per transaction. "Interactive" processing has the advantage of processing individual transactions quickly.

Some might argue there are exceptions to this rule, such as video games, graphics packages, screen savers, etc. Let's consider each individually: Video games write transactions to maintain player scores, graphics packages write a transaction every time you use the "Save/Cut/Copy/Paste" functions, and screen savers record preferred settings. All of these transactions must be written in specific formats. Understand this, if a program uses a clipboard or some other computer file, it is writing transactions to it. A program without any form of transaction serves no useful business purpose.

The tendency to be more concerned with the techniques of processing than with the problems to be solved can cause systems to be physically implemented in ways resulting in higher operating costs. The governing issues should be, *"What is the business problem to be solved?"* and, *"How effective is the processing method concerning the value and timing of the information to be produced?"* Programming techniques have little value in these areas. Consider, for example, that batch processing will always be a viable solution for information systems design regardless of the direction technology moves. This method of processing is analogous to many manufacturing operations where products are more economically manufactured in groups or batches - as opposed to one at a time. The point is, this situation will always be true regardless of how close the costs become between the two methods. For example, can you imagine preparing the corporate payroll "interactively" (one check at a time)? Technology should implement the logical business solutions to the problem and not be a platform to display technical elegance.

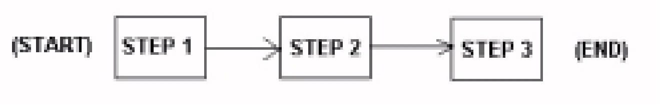
(continued on page 2)

(continued from page 1)

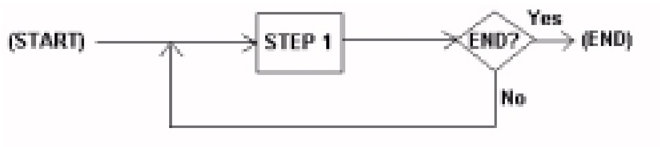
FORMS OF PROCESSING

There are three basic constructs for any processing, computer or otherwise (e.g., manual): sequence, iteration, and choice.

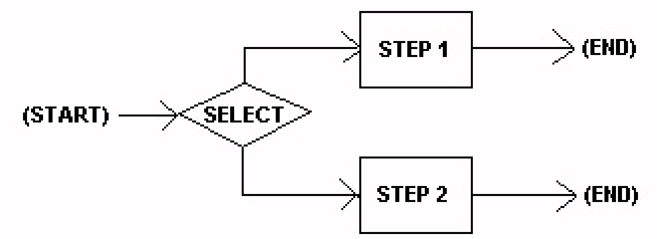
SEQUENCE - This type of processing represents a continuous series of steps, for example:



ITERATION - Represents repetition until a certain condition is met, for example:



CHOICE - Permits the selection of processing paths based on prescribed criteria. It also allows for parallelism; for example:



Obviously, these processing constructs can be combined in many different ways to process data. All procedures, programs, and steps consist of combinations of these constructs.

TIMING

As mentioned in the "PRIDE" Special Subject Bulletin, timing plays a substantial role in the design of a system, which is derived from the information requirements to be supported. Timing ultimately dictates when data is to be collected, stored, and retrieved. By using timing as a design parameter, we can synchronize the data base to serve all systems, not just one.

Timing (frequency, offset, and response time) dictates how often processing must occur, when each process is to start, and how fast it must process data. Without timing, processing will be awkward and cumbersome to perform. Since all business processes exist within a time frame anyway, why not use this as part of our design criteria from the outset as opposed to trying to synchronize the data base afterwards?

CONCLUSION

Transactions, processing constructs, and timing greatly influences the design of any process. For example, if a sub-system is needed upon request within a couple of seconds, then an "interactive" application is a likely solution. Conversely, if a sub-system is needed on a monthly basis to process voluminous transactions, then a "batch" process is a likelihood. However, if a sub-system is needed to process numerous transactions instantaneously, it may not be physically possible to do so (or practical).

All of this ultimately means the design of the process is ultimately based on the volume of transactions and the time required to process them. Buzzwords like "on-line," "real-time," "interactive," and "batch" only cloud the issue.

As an aside, it is interesting to see how such terminology has become a part of our vernacular. For example, the terms "on-line" and "off-line" really date back to the old days of computing. They were used to indicate whether a device was connected directly to a computer or not. As an example, the UNIVAC I had "off-line" card-to-tape and tape-to-printer devices. When the next generation was produced, the card reader and printer were connected directly "on-line" to the computer. Today "on-line/off-line" generally means if a person is logged on to a computer network or not.

But what is more important from a design point of view: our slick vocabulary or a true understanding of the fundamentals of processing? I think the latter. What do you think?

END

"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