

**TITLE: "BOXES AND LINES"**

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](mailto:timb001@phmainstreet.com)

Yahoo! IM: littleleaguerng

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

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

*"Documentation is an inherent part of the design process."*

*- Bryce's Law*

## INTRODUCTION

I recently overheard a Business Analyst say there was more to systems architecture than drawing boxes and arrows on a piece of paper. This may be true to a degree, but the ultimate deliverable of any engineering/architectural practice is a set of drawings from which to build a product. Architects and engineers do not spend all of their time drawing diagrams; for example, they have to specify requirements and analyze such things as the stress of components to determine the suitability of materials for use in design. But aside from this, the end result of engineering or architecture, their deliverable, is a set of drawings, be it a blueprint, a floor plan, wiring diagram, plumbing, or a set of flowcharts.

Such drawings basically consist of boxes and arrows. Boxes (be it squares, rectangles, polygons, circles, etc.) represent tangible objects and lines represent relationships between such objects. Flowcharts are similar; here, boxes represent specific types of processes or decisions or objects such as inputs/outputs/files, and lines represent dependencies between them (comes from/goes to).

Although drawings typically consist of geometric shapes, it is not uncommon to include tables or indices to represent decisions or to provide a cross-reference. Nonetheless, boxes and lines represent the principal means to visualize and communicate a design regardless of the structure to be built, and have been used since time immemorial.

In addition to diagramming techniques, engineers and architects have found it useful to develop models and prototypes to evaluate the overall physical aspects of their design. These are useful but let us not forget they are all ultimately based on a design of some kind (boxes

and lines). From the models and prototypes, designs can be adjusted as required.

I guess what I'm driving at is that despite all of this peripheral activity, and to refute my Business Analyst friend, the principal thrust of the engineer or architect is to produce and maintain a reliable set of drawings. It all comes down to boxes and lines. Interestingly, today's analysts and programmers think drawings are "old-hat" or passé. I don't care whether you draw it with pencil and paper or by computer, documentation is an inherent part of the design process. Failure to recognize this is to deny reality.

In terms of the Information Systems industry, flowcharts have been used for years, well before the introduction of the commercial computer in business. Originally they included process diagrams; later they were used by programmers as a convenient means to document program logic. Such flowcharts typically made use of ANSI standard flowcharting symbols. But as the Structured Programming movement flourished in the late 1970's, ANSI symbols were considered archaic, and many new types of diagramming techniques emerged, including Bubble Diagrams, Data Structure Diagrams, E/R Diagrams, HIPO, VTOC, etc. (anybody remember Nassi-Schneiderman Charts?). I could argue the pros and cons of the various techniques but that is not the point. What is important is that all of these diagramming techniques acknowledged documentation as an inherent part of the design process.

Today, documentation of any kind is considered a taboo (particularly among the Agile Methodology people). Small wonder the IT Industry is experiencing the same type of problems today that we experienced 35 years ago in terms of managing design complexity.

## BLUEPRINTING

It is a myth that one type of diagramming technique can be used for all development work. This would be like suggesting to use a wiring diagram to represent a floor plan. Different needs, different graphics, different purposes. There are actually four types of graphics to be used to the different levels of system design. This implies a blueprinting approach with various levels of abstraction, from general to specific. As we have discussed in the past, the "PRIDE"-Information Systems Engineering Methodology (ISEM) looks at a system as a product that can be engineered and manufactured like any other product and, as such, defines four levels of detail in a system's hierarchy:

*(continued on page 2)*

(continued from page 1)

**LEVEL 1**  
SYSTEM

**LEVEL 2**  
SUB-SYSTEMS (Business Processes)

**LEVEL 3**  
PROCEDURES (Administrative and Computer)

**LEVEL 4**  
PROGRAMS (for Computer Procedures)  
OPERATIONS (for Administrative Procedures)

Four different levels, four different graphics used:

**LEVEL 1**  
SYSTEM CONCEPT DIAGRAM - represents a freeform architectural rendering of the overall system.

**LEVEL 2**  
SYSTEM FLOWCHART - defines the SUB-SYSTEMS of the System.

**LEVEL 3**  
SUB-SYSTEM FLOWCHART - defines the PROCEDURES in a Sub-System (aka "Process Diagram").

**LEVEL 4**  
COMPUTER PROCEDURE FLOWCHART - defines the PROGRAMS in a Computer Procedure.

Each level provides the specifications for the next (this is also known as "stepwise refinement"). With the exception of the System Concept Diagram, all of the flowcharts make use of ANSI standard symbols. As to the internal processing logic of a program, since there are many ways to skin a cat, the software structure diagram du jour is used, hopefully a standard one. However, a graphic may not be necessary to express the processing logic of a program. Instead, specifications may be interpreted by a program generator of some kind. Its a "fielder's choice."

**CONCLUSION**

Until such time as we can master the Vulcan "mind meld," whereby we can transfer knowledge telepathically, there will always be a need for documentation. Its an inherent part of the design process and the principal deliverable produced by engineers and architects. Don't deny it, accept it.

I am definitely not one for excessive documentation

thereby becoming a burdensome task. Instead, documentation should be a natural byproduct of the design process. Just as blueprinting is an inherent part of the design process to architects and engineers, so should flowcharting be to system developers. And you shouldn't have to be a rocket scientist to draw a flowchart, keep it simple and try to use standard techniques for consistency instead of reinventing graphics every five minutes. As for me, I have no problem with ANSI standards; it works.

For additional information on graphics in systems design, consult the following sections from the "PRIDE" Methodologies for IRM:

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

Layered Documentation  
<http://www.phmainstreet.com/mba/pride/ismeth.htm#layered>

Graphics  
<http://www.phmainstreet.com/mba/pride/ismeth.htm#graphics>

**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 training and consulting on an international basis.*

"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/mbaprider/>

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

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

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

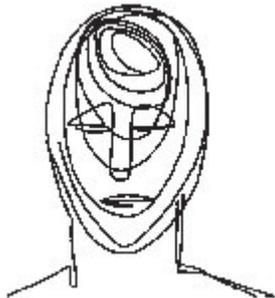
*Also, be sure to read Tim's Blog at:*

<http://blogs.ittoolbox.com/pm/irm/>

*"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 © MBA 2006. All rights reserved.



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