

FUNCTION ANALYSIS SYSTEMS TECHNIQUE – THE BASICS

FOREWORD

Some twenty years ago, J. Jerry Kaufman gave a presentation at the SAVE international conference describing the primary features of FAST and establishing today's accepted terminology¹ While SAVE International officially retains the rights to all Conference presentations, we gratefully appreciate Mr. Kaufman's continuing permission to reprint the portions of that presentation pertinent to this topic. The only material that follows not from that presentation are the editors notes and final summary.

INTRODUCTION

If you accept the premise that understanding the problem is 50 percent of the solution, then separating the problem from its symptoms and effects by analyzing its function is essential to the process.

The Function Analysis System Technique (FAST) is not an end product or result, but rather a beginning. It lays open the subject matter under study, forming the basis for a wide variety of subsequent study approaches and analysis techniques.

Developed by Mr. Charles W. Bytheway in 1964, and first presented as a paper to the Society of American Value Engineers conference in 1965, FAST contributed significantly to perhaps the most important phase of the Value Methodology (VM) – function analysis.

The VM discipline of describing a function by using a verb and noun is still intact. Distinguishing between basic and secondary functions and their subsets is also incorporated into the FAST process. The most dramatic differences are in the use of intuitive logic to “test” the functions and to graphically display them in a diagram or model form.

Function analysis is a common language, crossing all technologies. It allows multi-disciplined team members to contribute equally and to communicate with each other while addressing the problem objectively, without bias or preconceived conclusions.

In this presentation, the subject of FAST is explored from a software mode rather than the more common application to existing hardware. In fact, exposing the beginner to FAST through hardware examples may prove discouraging because the result and FAST diagram may not agree with the “obvious” structure or product under study. The tendency is then to “force fit” the FAST model to the product, instead of questioning the function relationship of the component parts. The fact that the hardware itself may not be the best way to achieve its required functions is not considered an option to the beginner, causing him or her to loose confidence and shy away from this most effective discipline.

As an effective management tool, FAST can be used in any situation that can be described functionally. However, FAST is not a panacea; it is a tool that has limitations which must be understood if it is to be properly and effectively used.

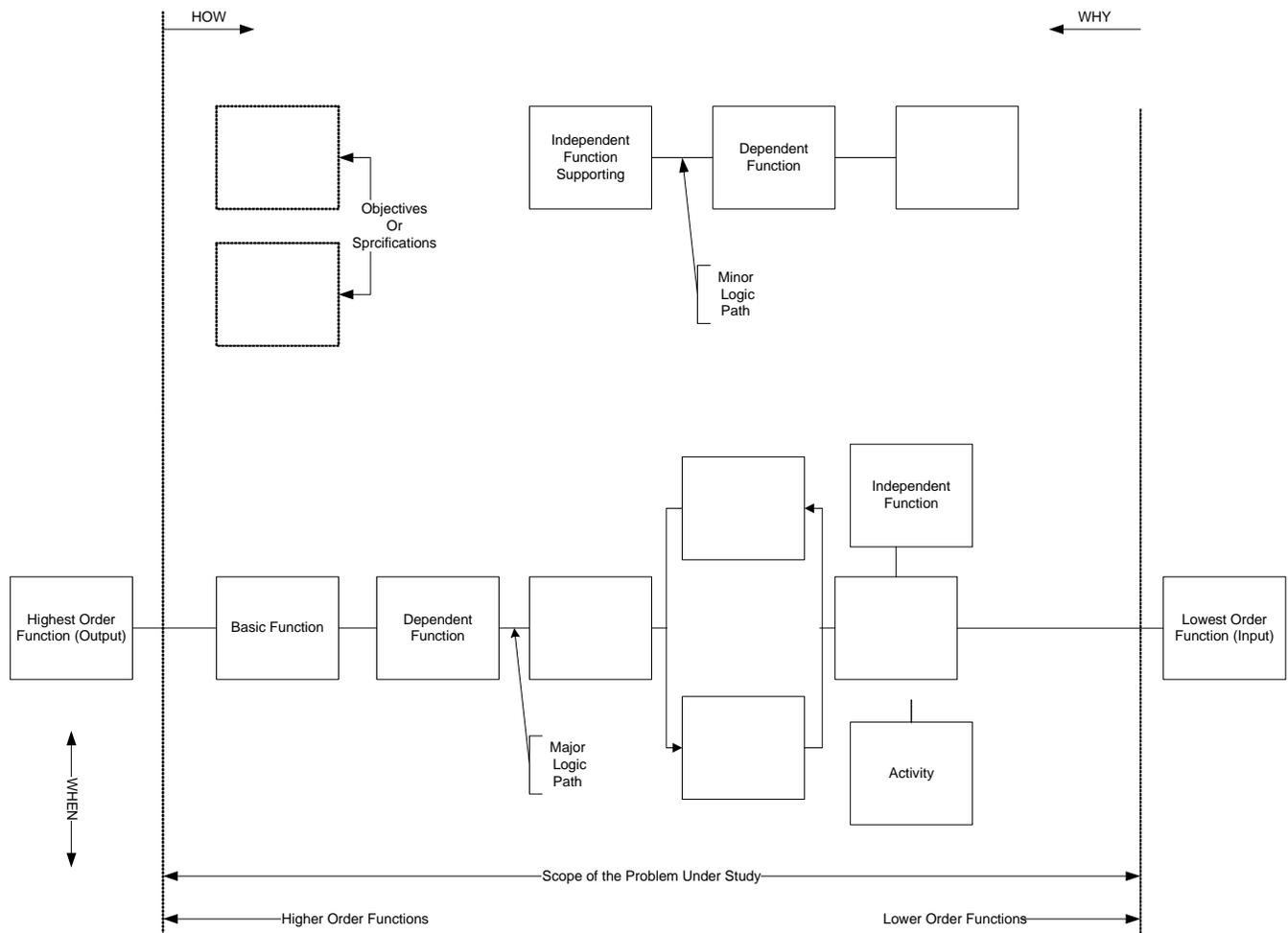
FAST is a system without dimensions: That is, it will display functions in a logical sequence, prioritize them and test the dependency; but, it will *not* tell you *how well* a function should be performed (specification), *when* (not time oriented), *by whom*, or *for how much*. (Editor’s note: However, these dimensions can be added to the model. Which dimensions to use is dependent on the objectives of the project. This feature as will be discussed in the monograph, FAST- Advanced Applications).

There is no “correct” FAST model as in comparing it to some text solution, but there is a “valid” FAST model. Its degree of validity is directly dependent on the talents of the participating team members, and the scope of the related disciplines they can bring to bear on the problem. The single most important output of the multi-disciplined team engaged in a FAST exercise is *consensus*. There can be no minority report. FAST is not complete until the model has the consensus of the participating team members and adequately reflects their input.

THE MECHANICS OF THE PROCESS

To begin the process, a basic model is offered. (Ref. Figure A) showing the FAST components and describing their parts.

Figure A – The Basic FAST Model



A. *Scope of the problem under study*

Depicted as two vertical dotted lines, the scope lines bound the problem under study, or that aspect of the problem with which the study team is involved.

B. *Highest order function(s)*

The objective or output of the basic function(s) and subject under study, is referred to as the highest order functions, and appears outside the left scope line, and to the left of the basic functions.

Note: Any function to the left of another on the critical path is a “higher” order function.

C. *Lowest order function*

These functions to the right and outside of the right scope line represent the input side that “turns on” or initiates the subject under study and are known as lowest order functions. Any function to the right of another function on the critical path^A is a “lower” order function.

Note: The term “higher” or “Lower” order functions should not be interpreted as relative importance, but rather the input and output side of the process. As an example, “receiving objectives” could be the lowest order function with “satisfying objectives” being the highest order function. How to accomplish the “satisfy objectives” (highest order function) is therefore the scope of the problem under study.

D. *Basic function(s)*

Those function(s) to the immediate right of the left scope line represent the purpose or mission of the subject under study.

E. *Concept*

All functions to the *right* of the basic function(s) describe the approach elected to achieve the basic function(s). The “concept” represents either existing conditions or a proposed approach. Which approach to use (current or proposed) is determined by the task team and the nature of the problem under study.

F. *Objective or specifications*

Objectives or specifications are particular parameters or restrictions which must be achieved to satisfy the highest order function in its operating environment. Although they are not themselves functions, they may influence the concept selected to best achieve the basic function(s) and to satisfy the users’ requirements.

G. *Critical path function(s)^A*

Any function on the *How* or *Why* logic is a critical path function. If a function along the *Why* direction enters the basic function(s), then it is a *major* critical path; otherwise, it will conclude in an independent (supporting) function and be a *minor* critical path. Supporting functions are usually secondary. They exist to achieve the performance levels specified in the objectives or specifications of the basic functions, or because a particular approach was chosen to implement the basic function.

Note: Independent functions (above the critical path) and activities (below the critical path) are the result of satisfying the *When* question.

H. *Dependent functions*

Starting with the first function to the right of the basic function, each successive function is “dependent” on the one to its immediate left, or higher order function, for its existence. That dependency becomes more evident when the *How* question and direction are followed.

I. *Independent (or supporting) function(s)*

Functions that do not depend on another function, or method selected to perform that function. Independent functions are located above the critical path function(s), and are considered secondary, with respect to the scope, nature, and level of the problem, and its critical path.

J. *Function*

An end or purpose that a “thing” or activity is intended to perform, expressed in a verb-noun form.

K. *Activity*

The method selected to perform a function (or group of functions).

(Editor’s note: In recent years, the activities are not normally shown on the FAST diagram, but rather used in the analysis to determine when to stop listing functions: i.e., when defining functions the next connection is an activity, and then the team has defined the functions to their lowest level. Therefore, today’s teams place the independent functions both above and below the major critical path.)

To those who are system oriented, it would appear that the FAST diagram is constructed backwards, because in systems terms (or the *Why* direction) the “input” is normally to the left side, and the “output” to the right. However, note that when a method to perform a function on the critical path is changed, it affects all functions to the *right* of the changed function, or stating it in function analysis terms, changing a function will alter the functions *dependent* upon it. Therefore, the *How* (reading left to right) and *Why* (reading right to left) directions are properly oriented in FAST logic.

Editor’s note: The monograph “Function Relationships - An Overview #2” describes the development of a second model, referred to as Technical FAST. That model—and procedure separates out “all the time function(s)” and “one time function(s)” since these functions often required forced fitting into the above model. Otherwise, terminology, logic, etc. were retained.

STANDARD SYMBOLS AND GRAPHICS

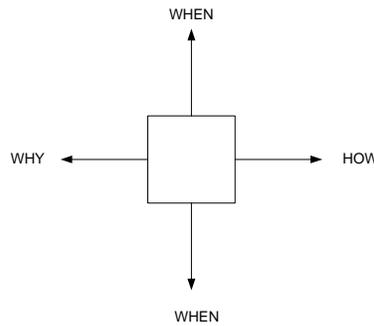
Most examples of FAST diagrams and models show variations in graphically representing the problem. Although the task teams working the exercise understand the notations, it is difficult for outsiders, even those familiar with the FAST technique, to read the model without the aid of a participating team member.

Even within the same company, the need for a uniform graphic language becomes apparent when executive management asks to review the FAST outputs from a number of section managers describing the current or proposed functions of their departments.

IF FAST IS TO BE UNIVERSALLY ACCEPTED, IT MUST FIRST BE UNIVERSALLY UNDERSTOOD

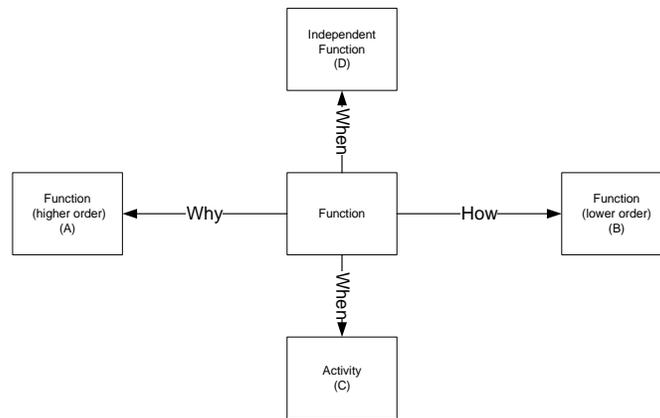
The following notations are offered as a recommendation for standardizing the symbols and graphics developed after some thirty-odd task assignments. There have been a number of alternatives in trying to adapt a convention that could be quickly understood, yet allow graphic flexibility to expand the model when needed. Why certain symbols were adopted will be explained where it is thought appropriate. However, the importance is in standardizing a common language, not in debating which symbol is used where. (Editors note: These symbols and graphics have become the accepted standard over the past 20 years.)

The four (4) primary directions in a FAST diagram are:



The **HOW** and **WHY** directions are always along the critical path, whether it be a major or minor critical path. The **WHEN** direction indicates an independent or supporting function (up) or activity (down). At this point, the rule of always reading from the exit direction of a function in question should be adopted so that the three primary questions **HOW**, **WHY** and **WHEN** are answered in the blocks indicated below.

DETERMINATE LOGIC



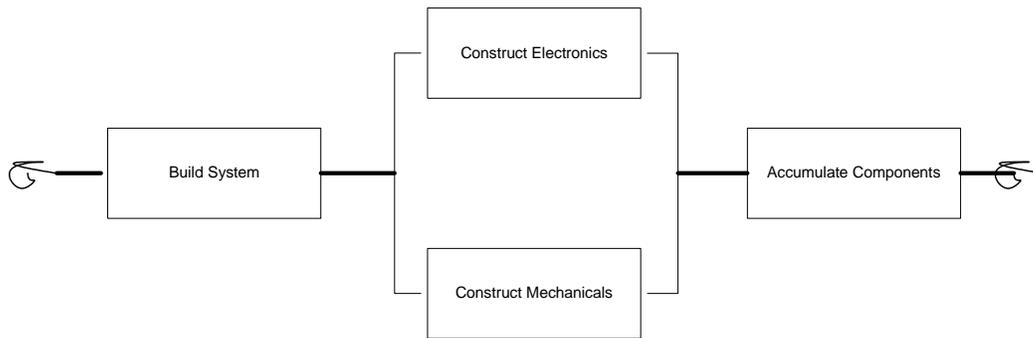
1. **HOW** is (function) to be accomplished? By (B)
2. **WHY** is it necessary to (function)? So you can (A)
3. **WHEN** (function) occurs, what else happens? (C) or (D)

The answers to the three questions above are singular, but they can be multiple (AND), or optional (OR). Also their relative importance may be noted.

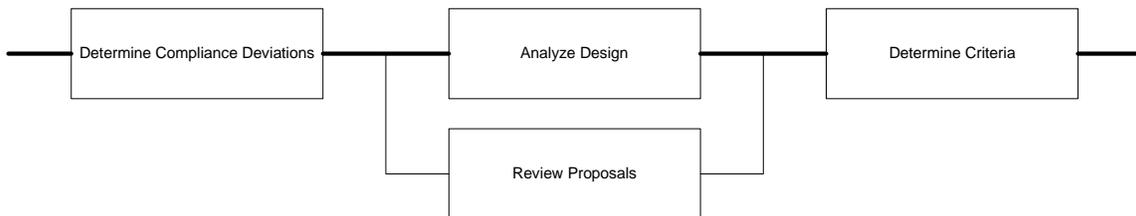
ALONG THE CRITICAL PATH - AND

“AND” is represented by a split or fork in the critical path. In both examples the fork is read as “AND.” In Example A, How do you “Build System”? By “Constructing Electronics)” AND “Constructing Mechanicals.” In Example B, how do you “Determine Compliance Deviations?” By “Analyzing Design” AND “Reviewing Proposals.” However, the way the split is drawn, Example A shows “Constructing Electronics” and “Constructing Mechanicals” equally important; and in Example B, ”Analyzing Designs” is shown as being considered *more important* than “Review Proposals.”

EXAMPLE A
“AND” (Equally Important)



EXAMPLE B
“AND” (Less Important)

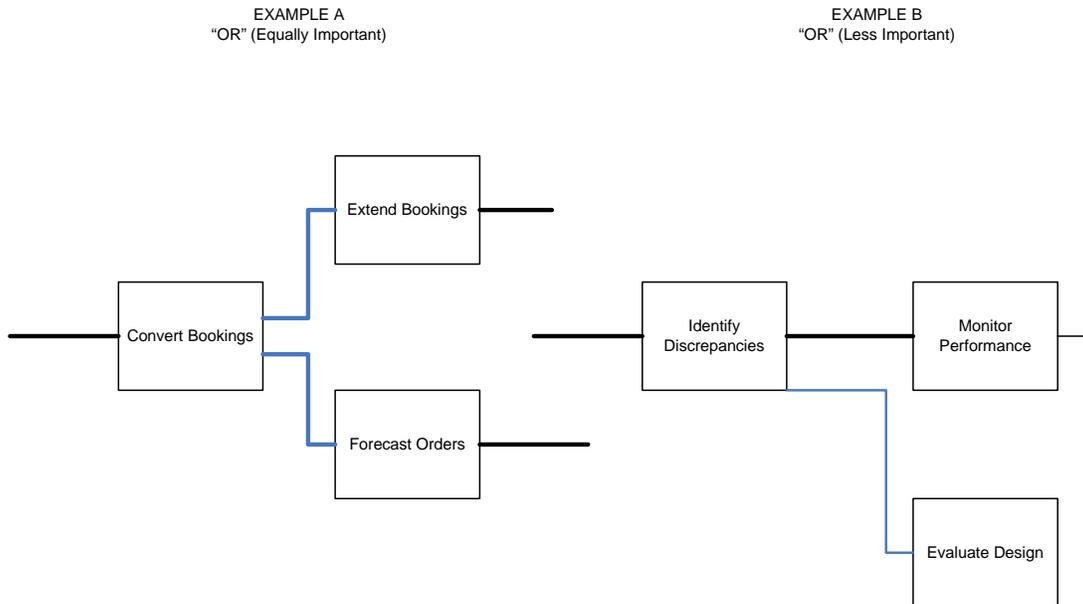


ALONG THE CRITICAL PATH - “OR”

“Or” - represented is represented by multiple exit lines indicating a choice.

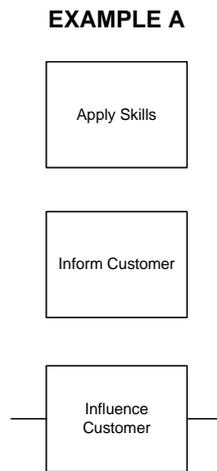
Using Example A, the answer to the question, how do you “convert bookings,” is by “Extending bookings” **OR** “Forecast Orders.” When going in the *Why* direction one path at a time is followed. Therefore: “Why do we “Extend Bookings?” So you can “Convert Bookings.” Also, why do you “Forecast Orders?” So you can “Convert Bookings to Delivery.”

The same process applies to Example B, except as in the AND example: “Evaluate Design” is noted as being less important than “Monitor Performance.”



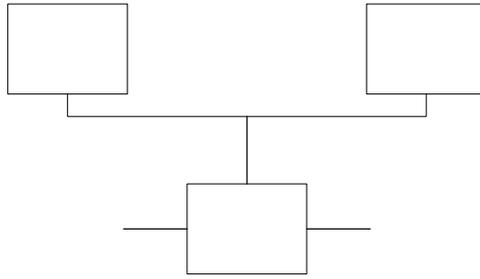
“AND” Along the WHEN Direction

WHEN functions (applicable to independent functions and activities) AND is indicated by connecting boxes above and/or below the critical path functions.

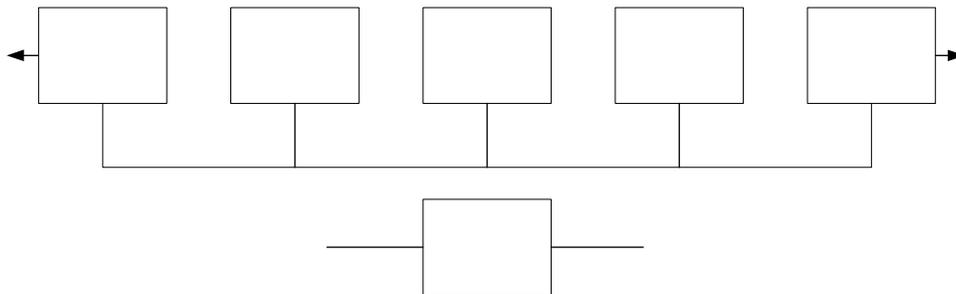


The above example states “When you influence the customer,” you “Inform the customer” AND “apply skills.” If it is necessary to rank or prioritize the AND functions, those closest to the critical path function should be the most important.

As a note of explanation. It would appear that the same “fork” symbol should be used to express AND in the WHEN as well as the HOW-WHY direction, giving example A this appearance.



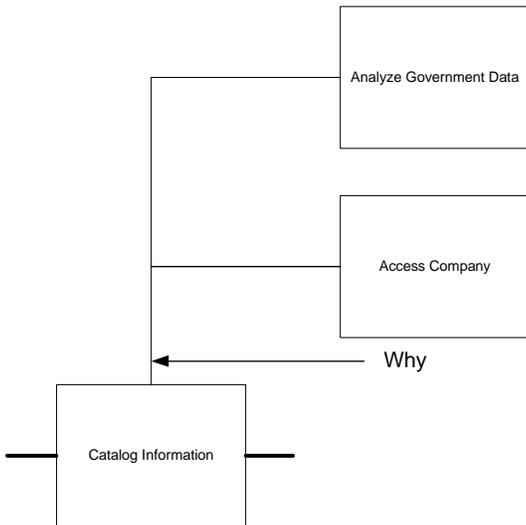
However, to do so would cause some graphic problems in multiple AND functions, in addition to building minor critical paths, such as:



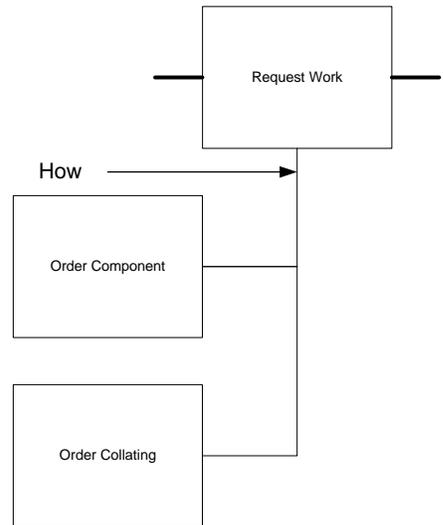
“OR” Along the WHEN Direction

OR is indicated by “flags” facing right or left.

EXAMPLE C (WHY)



EXAMPLE C



Although Example B is an independent function (above the critical path) and Example C is activities (below the critical path), the WHEN, #OR rules are equally applicable to both. Locating the “flags” to the left or right of the vertical line bears on how you read back into the critical path function.

In example B, reading from the critical path function, when you “catalogue information,” you “assess company data” OR “analyze government data” (generally not both). However, in reading back from the function, the exit lines are WHY oriented.

Why do you “assess company data” OR “analyze government data?” So you can “catalogue information”.

In Example C, working from below the critical path function, it reads: When you “request work,” you “order components” OR “order collating.” Since the blocks are below the critical path, they are activities. In reading activities back, the flags face the HOW direction and the question reads: HOW do you “order components” OR “order collating?” By “requesting work.”

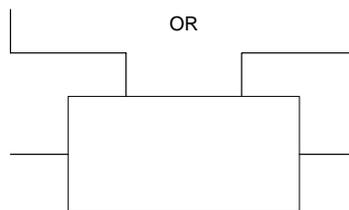
Notice that the flag is better suited for this example than if the flags faced the WHY direction.

Once again the graphic considerations modified the OR notations as seen on the critical path. Since OR is expressed in this form on the critical path:



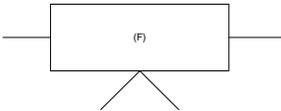
It would appear that OR in the WHEN direction should follow this convention.

However, the same problem in building minor critical paths from the support functions would occur. Also, the “flag” OR, reading back into the critical path, would be more difficult to express.



OTHER NOTATIONS AND SYMBOLS

Other notations and symbols used in expressing ideas and thoughts in the FAST Model are as follows:

- 
Indicates that the network continues, but is of no interest to the team, or does not contribute to the solution sought.
- 
Indicates the Function (F) is further expanded on a lower level of abstraction.
- 
Indicates that the line X connects elsewhere on the model, matching the same number.

SUMMARY (By editor)

This presentation and the materials it contains have remarkably withstood the test of time over twenty years and thousands of FAST networking by a host of value study teams. The only principal change is that rarely today do we see the use of the “activity” designation. To avoid confusion and enhance communication with the product/project users, all elements on the FAST diagram are functions. Activities are more commonly used to determine when to stop the function analysis process: i.e., When the next logical step is an activity (the method or task to perform a function), the team concludes that is the end of defining functions in that chain. However, the original paper is included in this monograph both for historical purposes and for those who wish to use the more sophisticated approach.

The rest of the language and symbols have become standard usage. As Mr. Kaufman stated, “If FAST is to be universally accepted, it must first be understood” through the international usage of common language.

In May 1997, SAVE-International issued its first Value Methodology Standard. Many of the definitions in the glossary of that document concern FAST diagrams. They are presented below for currency, but the reader can easily ascertain that the intent of SAVE International’s current terminology and those in Mr. Kaufman’s paper have the same intent.

^A In reviewing this monograph Mr. Kaufman suggested that the newer and more meaningful term for the “critical” path is “Logic” path. That also reduces confusion for those new to FAST methodology but who are familiar with Program Evaluation Review Technique (PERT) and/or Critical Path Methodology (CPM). However, virtually all current SAVE International literature, including the Certification Board’s Body of Knowledge, still use the “Critical Path” designation.

¹ Glossary from SAVE International Value Methodology Standard, May 1997, Revised October 1998, (selected definitions).

FUNCTION

The natural or characteristic action performed by a product or service.

FUNCTION, BASIC

The primary purpose or most important action performed by a product or service. The basic function must always exist, although methods to achieve it may vary.

FUNCTION ANALYSIS SYSTEMS TECHNIQUE (FAST)

A horizontal chart depicting functions within a project, with the following rules:

- a. The sequence of functions on the critical path proceeding from left to right answers the question “How is the function to its immediate left performed?”
- b. The sequence of functions on the critical path proceeding from right to left answers the question “Why is the next function performed?”
- c. Functions occurring at the same time or caused by functions on the critical path appear vertically below the critical path
- d. The basic function of the study is always farthest to the left of the diagram of all functions within the scope of the study.

e. Two other functions are classified:

- 1) Highest Order - The reason or purpose that the basic function exists. It answers the “why” question of the basic function and is depicted immediately outside the study scope to the left.
- 2) Lowest Order - The function that is required to initiate the project and is depicted farthest to the right, outside the study scope. For example, if the value study concerns an electrical device, the “supply power” function at the electrical connection would be the lowest order function.

SCOPE:

The portion of the overall project that is selected for the value study. The analysis accepts everything within the defined scope in order to focus attention on the functions within those limits.