# A Primer For Model-Based Systems Engineering

2nd Edition

David Long
Zane Scott

Vitech

Copyright © 2011 Vitech Corporation.  All rights reserved.

ISBN 978-1-105-58810-5 (paperback edition)

Permission to reproduce and use this document or parts thereof and to prepare derivative works from this document is granted, provided that both attribution to Vitech Corporation and this copyright notice are included with all reproductions and derivative works.

Product names mentioned herein are used for identification purposes only, and may be trademarks of their respective companies.

Publication Date:  October 2011

# CONTENTS

THIS PAGE INTENTIONALLY BLANK

# INTRODUCTION

This is the 2nd edition of Vitech's model-based systems engineering primer. In this second treatment of the subject, we have covered the same subject matter as before but augmented this time with what we have learned since releasing the 1st edition. We strive to be a "learning organization" and to leverage that learning for the benefit of our customers and community. With this edition we hope to carry that principle forward.

There are notable differences in this edition. First, we have reorganized the material. Instead of the topical organization of the 1st edition, we have approached the description of model-based systems engineering (MBSE) from a "building blocks" perspective suggested by its name. We ask first "What Is a System?" From there we tackle "What Is Systems Engineering?" Then we discuss "What Is a Model?" and finally arrive at the question "What Is Model-Based Systems Engineering?" We hope that this building approach will make it easier to put the concepts into a logical framework for understanding and use.

We have also tied the concept discussions more closely to practical illustrations. We have largely drawn these from the example system design included with this primer. This has been done in response to many helpful suggestions from our readers, and we think it makes the concepts much clearer and easily understood.

One of the most common flaws in any undertaking is a departure from the fundamental principles of the disciplines involved in the process. This can be due to inattention bred by familiarity or a failure to recognize and reinforce "the basics." Whether the enterprise is a football game or a systems design project, the fundamentals of "blocking and tackling" are critical to success. Absent or poorly executed, they can doom the venture. In the case of a floundering effort, they are the key to getting back on track.

The importance of knowing and executing the basics is the driving force behind this primer. It is the reason for not beginning with a collection of essays on more advanced topics. Revisiting the "blocking and tackling" aspects of MBSE is the foundation of our effort to advance the cause of sound systems design.

This primer addresses the basic concepts of model-based systems engineering. It covers the Model, Language, Behavior, Process, Architecture, and Verification and Validation. It is a call to consider the foundational principles behind those concepts. It is not designed to present novel insights into MBSE so much as to provide a guided tour of the touchstones of systems design. It is a guide to the new MBSE acolyte and a reminder to the experienced practitioner.

Why such a basic approach? Without this grounding, it can become easy to lose the sense of relationship between techniques and the design itself. Reading and pondering the sections on Models and Language bring into focus the difference between representations of the model and the model itself. A map may be an extensive, informative, and important representation of the underlying terrain, but it is simply that—a representation. Likewise, a set of diagrams may be useful, clear, and detailed, but they are not the model of the system itself. Without returning to the concepts of model, language, process, and behavior, we can easily become mistakenly convinced that the process of drafting a "full set" of representations is the same thing as constructing a model. It is through understanding the basics that we understand the distinctions.

In other ways as well, the failure to be aware and alert to the basic principles of MBSE can hinder the integrity of a system-design effort. Just as a football team returns again and again to their roots in basic skills, we can all profit from reacquainting ourselves with these basic principles.

This primer is offered to the end that it will function as a call back to the basic concepts of our discipline. It lays the groundwork for improvements and enhancements already being planned. As it stands, it is a look at the foundational concepts of MBSE designed to benefit the newcomer and experienced practitioner alike.

Finally, for whom is this primer intended? Of course the obvious answer is that it is intended to be an introduction to these concepts for those who may be new to the world of model-based systems engineering. It is written in a way that can be understood by any intelligent and curious reader—even if that reader is not an engineer. Project managers, acquisitions professionals, and business-process consultants can all use this primer to guide them into the MBSE concepts in an organized way.

In addition, this primer is intended to provide an organized presentation of these concepts for the systems engineering practitioner who may need a reference framework for them. Often we become familiar with the concepts we use in the way that we customarily use them. This is as true in the systems engineering discipline as in any other. Like a gradually fragmenting hard drive, our thinking becomes compartmentalized in ways that reflect how much and when we use the concepts we have learned over time.

With this primer we hope to provide the seasoned practitioner a framework in which to refresh the concepts and see the relationships. Any experienced systems engineer can take these and expand them with additional detail and application anecdotes. It is our hope that this basic discussion can provide a "rack" in which to place that experience and, thereby, make it more useful in practice.

It is our hope that you find this primer valuable. We welcome your comments and suggestions about improving it. Much of what we have learned about how it should be organized and presented has come from thoughtful contributions from the readers of the 1st edition.

***Vitech Corporation***

***October 2011***

# WHAT IS A SYSTEM?

Although the term *system* is defined in a variety of ways in the systems engineering community, most definitions are similar to the one used in the U.S. Department of Defense Architecture Framework (DoDAF)—"any organized assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions." (*Department of Defense Dictionary of Military and Associated Terms,* its.bldrdoc.gov/fs-1037/dir-036/_5255.htm).

In her book *Thinking in Systems: A Primer* (Chelsea Green Publishing, 2008), Donella Meadows puts it somewhat more succinctly by saying "A system is an interconnected set of elements that is coherently organized in a way that achieves something." She goes on to point out that "A system is more than the sum of its parts. It may exhibit adaptive, dynamic, goal-seeking, self-preserving, and sometimes evolutionary behavior."

The idea that a system is "more than the sum of its parts" is picked up in the International Council on Systems Engineering (INCOSE) definition of a system. INCOSE defines a system as "a construct or collection of different entities that together produce results not obtainable by the entities alone" ("A Consensus of the INCOSE Fellows", www.incose.org/practice/fellowsconsensus.aspx).

There are some clear commonalities among these three definitions. First, any system must be made up of what Meadows refers to as "elements" (called an "assembly of resources and procedures" in the DoDAF definition and "entities" by INCOSE). These are the parts of the system that together form the whole.

In addition, a system must tie the parts together with relationships. What Meadows calls “interconnect(ion)” and “coherently organized” is for INCOSE a “construct” and for DoDAF an “organized assembly . . . united and regulated by interaction or interdependence.” The concept of relationships is the second common characteristic of these definitions.

Finally, the system must have a purpose for which the elements are assembled. DoDAF calls this organizing purpose “to accomplish a set of specific functions.” INCOSE sees it as the ability to “produce results not obtainable by the entities alone.” For Meadows the system is “organized in a way that achieves something.” In each definition the elements of the system are related to each other in ways that promote the accomplishment of a specific purpose that is beyond the capability of any of the parts acting alone.

It should be noted at this point that these aspects of a system are often construed narrowly in practice, causing our view of systems to be constrained or limited. In reality, systems exist wherever these three are present: parts, relationships, and a purpose. This primer will be intentionally broad in its view of where those aspects are present. This will enable us to see systems in places where heretofore we might not have expected them.

Some examples of systems include:

- A set of things working together as parts of a mechanism or an interconnecting network.
- A set of organs in the body with a common structure or function.
- A group of related hardware units or software programs or both, especially when dedicated to a single application.
- A major range of strata that corresponds to a period in time, subdivided into series.

- A group of celestial objects connected by their mutual attractive forces, especially moving in orbits about a center.

The entities or elements of a system constructed by humans can include people, hardware, software, facilities, policies, and documents. For any given system, this list is limited only by the set of things required to produce its system-level results. These results include system-level qualities, properties, characteristics, functions, behavior, and performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationships among the parts. In other words, the "value-add" of the system emerges in the synergy created when the parts come together.

The sample problem included with this primer is that of a Geospatial Library tasked with connecting a set of satellite imaging collectors with its customers. It is composed of parts. At the highest level, it has a Command Center Subsystem that manages the collection, storage, and retrieval of imagery products as well as a Workstation Subsystem that manages the translation of various incoming imagery requests into an internal common imagery collection request.

The system parts have relationships that define their interaction and the system's function. For example, when a customer request cannot be serviced from the system's current inventory, the Command Center Subsystem makes a specific collection request to a specific sensor to satisfy the customer's need. The products generated from this request are collected, added to the Geospatial Library inventory, and combined into a package for shipment to the customer by the Workstation Subsystem. Together, the parts of the system have allowed the system to take a specific customer request, obtain the images necessary to satisfy that request, and provide the images to the customer in response.

This fulfills the system's purpose—servicing the needs of the customers and collectors in facilitating the exchange of requests for images and the images themselves. At a very high level this illustrates the presence of the parts, relationships, and purpose in the sample system.

## Systems Thinking

In order to create systems, it is necessary to engage in "systems thinking." Peter Senge's book *The Fifth Discipline* (Doubleday/Currency, 1990) introduced systems thinking into popular culture. However, it remains largely unappreciated and is honored mainly in the breach rather than the observance. Part of this is because systems thinking is practiced using too narrow a definition of "systems," and this narrowness limits the practice of systems thinking.

For a broad understanding of what is meant by "systems thinking," we will turn to one of the preeminent systems thinkers, Russell Ackoff. Notice his use of the three aspects of the systems definition (parts, relationships, and purpose) in defining systems thinking: "systems thinking looks at relationships (rather than unrelated objects), connectedness, process (rather than structure), the whole (rather than just its parts), the patterns (rather than the contents) of a system, and context" (R. Ackoff with H. Addison and A. Carey, *Systems Thinking for Curious Managers*, Triarchy Press, 2010).

Ackoff goes on to state, "Thinking systemically also requires several shifts in perception, which lead in turn to different ways to teach, and different ways to organize society." This statement is significant in two ways. First, Ackoff is observing that the move to systems thinking requires changing the way we think. In addition, he is showing that he sees systems (and systems thinking) quite broadly.

Taking his latter suggestion first, Ackoff is interested in the application of systems thinking beyond the classic boundaries of systems engineering. Coming from a business and process orientation (as opposed to an engineering orientation), Ackoff sees the concepts of systems and systems thinking as broadly applicable to business and even social process design. In his book *Redesigning Society* (R. Ackoff and S. Rovin, Stanford Business Books, 2003), he focuses on the systems aspects of public policy decision making. He is truly committed to the idea of seeing systems wherever the three aspects are present.

Perhaps his most important insight has to do with the "shifts in perception" or changes in thinking involved in thinking systemically. A major shift in thinking comes from moving away from the exclusively analytic approach that has characterized our thinking since the Enlightenment. This analytic approach, according to Ackoff, "is a three-step process: (1) take the thing or event to be understood apart; (2) explain the behavior or properties of the parts taken separately; and (3) aggregate the explanations of the parts into an understanding of the whole, the thing to be explained" (Ackoff and Rovin, Redesigning Society). Such "analytic thinking" takes our focus off of the system and orients it to the parts individually. This analytic, parts-oriented approach leads too often to ill-fated attempts to improve system performance by improving the parts of the system. Not only are such attempts typically fruitless, but they can actually damage overall system performance or even destroy the system.

What is needed is a different way of thinking, a way of approaching problems from a systems perspective. Ackoff calls this new approach "synthetic thinking." According to Ackoff, "Synthetic thinking is also a three-step process, each the opposite of the corresponding step of analysis: (1) identify one or more systems that contain the system to be explained; (2) explain the behavior of the containing system (or systems); and (3) disaggregate the understanding of the containing system into the role or function of the system to be explained" (Ackoff and Rovin, Redesigning Society). The critical idea here is that we begin not from a decomposition of the system into its parts but from the point of view of the system in its context.

In the book *Systems Thinking for Curious Managers*, Ackoff points out that "Managers should never accept the output of a technologically-based support system unless they understand exactly what the system does and why. Many managers who are unwilling to accept advice or support from subordinates whose activities they do not fully understand, are nevertheless willing to accept support from computer-based systems of whose operations they are completely ignorant. Management information systems are usually designed by technologists who understand neither management nor the difference between data and information. Combine such ignorance with a management that does not understand the system the technologists have designed, and one has a recipe for disaster or, if lucky, large expenditures that bring no return" (Ackoff, Russell; Addison, Herbert; Carey, Andrew; Gharajedaghi, Jamshid (2010). *Systems Thinking for Curious Managers: With 40 New Management f-Laws*).

The point here is that systems must be understood in the context of what they can do and the world in which they will do it. It is not enough to see the system as a sum of the operations of the component functions. It must been seen as a functioning whole. This is the systems viewpoint.

This viewpoint allows us to engage the system without losing sight of the context and purpose of the system as a whole. Effective systems thinking combines analytic and synthetic thinking. It is common to see analytic thinking without its synthetic sibling. Too often this results in the loss of systems perspective. At its worst, this becomes component engineering.

The loss of the systems perspective can be quite costly. When the consequences of a limited or missing systems view emerge during the design process—as when different design paths result in mutually exclusive constraints—the penalty is expensive rework. Cost and schedule suffer together as the system is reengineered to correct the problems.

Sometimes the missing perspective doesn't levy its price until the system is built. This is the failure that Ackoff calls out—the failure to "understand exactly what the system does and why." This leads easily to unintended consequences as the system interacts with its environment in unanticipated and unhelpful ways.

## Applying Systems Thinking to Systems Design

A system begins with an idea that must be translated into reality. The theoretical idea of a system must link to the engineered system "reality" and vice versa (bidirectional linkage). The designers must also find a way to clearly show when and how the theory explains reality and how reality confirms their theory.

The system design must take into account the system properties. Within the boundary of a system, there are three kinds of properties:

**Entities**—These are the parts (things or substances) that make up a system. These parts may be atoms or molecules; larger bodies of matter like sand grains, raindrops, plants, or animals; or even components like motors, planes, missiles, etc.

**Attributes**—Attributes are characteristics of the entities that may be perceived and measured such as quantity, size, color, volume, temperature, reliability, maintainability, and mass.

**Relationships**—Relationships are the associations that occur between entities and attributes. These associations are based on cause and effect.

In order to explain the design, the engineers must use some form of expression. When taken together, the properties of the system—the entities, attributes, and relationships—form a system "language." This language is fundamental to being able to describe and communicate the system among the engineering team as well as to other stakeholders.

Using this language, the system can be represented hierarchically, allowing it to be understood as decomposable into meaningful subunits. These subunits are conventionally named:

- a *system* is composed of subsystems;
- *subsystems* in turn are composed of assemblies;
- *assemblies* are composed of subassemblies, and
- *subassemblies* are composed of parts.

It is important to note that what may be considered a "part" in the context of a particular system may be a complete "system" in its own right. This all depends upon the point from which the system is viewed and the resulting system boundary decisions.

Often the terms used in describing this hierarchy are not well specified; some engineers use the term *sub-subsystem*, others use the terms *component* and *subcomponent* in the hierarchy. Variant usage only contributes to confusion. In order to avoid such usage confusion, the term *component* is used here as an abstract term representing the physical or logical entity that performs a specific function or functions.

The parts of a system interact to produce the performance of the whole system. It is intuitively obvious that all parts of the system must be functioning as designed in order for the system to function properly. What is not so obvious is that improving the function of one of the parts, be that a subsystem or component or whatever it may be labeled, will not necessarily improve the functioning of the whole. This is because of the effects of interaction within the system. For example, improving the resolution of the images gathered by the collectors in the sample problem will not improve the product for the customer if the image inventory cannot process and deliver them. Any improvement must be considered from a perspective that looks across the system as a whole.

The system results at the customer level depend upon the performance of the entire system. While the components must be understood from the perspective of whether or not they can perform the behavior allocated to them by the system design, it is ultimately the performance of the system that matters. This must account not only for the capability to meet the needs of the stakeholders that drove the system creation but also for any extraneous consequences of system performance, particularly unintended or unplanned consequences. Understanding and practicing this is the very foundation of systems thinking.

THIS PAGE INTENTIONALLY BLANK

# AUTHORS

David Long founded Vitech Corporation in 1992 to develop and commercialize CORE®, a leading systems engineering software environment used globally. He continues to lead Vitech in delivering innovative solutions and empowering organizations to develop and deploy next-generation systems.

For over twenty years, David has focused on enabling, applying, and advancing model-based systems engineering (MBSE) to help transform the state of the systems engineering practice. He has played a key technical and management role in refining and extending MBSE to expand the analysis and communication toolkit available to systems practitioners. David is a frequent presenter at industry events worldwide delivering keynotes and tutorials spanning introductory systems engineering, the advanced application of MBSE, and the future of systems engineering.

Zane Scott manages Vitech's Professional Services and Training organization. For the past twenty-five years, Zane has built a skill set which enables him to provide insight and guidance to individuals and companies as they improve organizational processes and methodologies. Zane has also taught systems engineering methodology in the practical process context in a variety of settings.

Zane brings a unique perspective to Vitech and its customers. With a professional background in the litigation field, Zane is also a trained negotiator, labor management facilitator and mediator. He has practiced tactical negotiation and interventional mediation and taught communications, conflict management and leadership skills at both the university and the professional level. Before joining Vitech, Zane worked as a senior consultant and process analyst assisting government and industry clients in implementing and managing organizational change.

**Vitech Corporation**

2270 Kraft Drive, Suite 1600
Blacksburg, Virginia 24060
540.951.3322  FAX: 540.951.8222
www.vitechcorp.com
community.vitechcorp.com