TWELVE SYSTEMS ENGINEERING ROLES

Sarah A. Sheard
Software Productivity Consortium
2214 Rock Hill Rd, Herndon VA 22070
sheard@software.org, (703) 742-7106

Abstract. Twelve roles are described which are occasionally or frequently assumed to constitute the practice of systems engineering. Some roles fit naturally as life-cycle roles, others fit the Program Management set of roles, while still others are not normally thought of in either group. Interactions between the roles are discussed, and the systems engineering roles assumed by the papers in the inaugural issue of *Systems Engineering*, the Journal of INCOSE, are compared to these categories.

INTRODUCTION

Since its inception, INCOSE has been attempting to resolve the question of what, exactly, is systems engineering. Several dualities have been explored, including whether systems engineers are specialists or generalists, and whether systems engineering is a set of life-cycle roles, such as the generation of specifications and verification programs, or an overall program management discipline. There has even been a discussion on whether systems engineering is a discipline or an attitude [Mar 92]. Worthy and wise arguments have been put forth on both sides of each issue, leaving some to despair of ever being able to pin down definitions that all can agree on.

A local chapter presentation on the value of systems engineering1 provided the impetus for this paper. The presenters seemed to be talking about entirely different definitions of systems engineering and the roles that systems engineers play. A compilation of the roles seemed essential to deciding many important questions in the field of systems engineering. A companion paper in this volume, "The Value of Twelve Systems Engineering Roles" [Sheard 96], addresses the value of systems engineering from the point of view of the roles described in this paper.

To derive these twelve systems engineering roles, papers in the inaugural issue of *Systems Engineering*, the Journal of INCOSE, were reviewed for assumptions about roles that systems engineers play. More than sixty descriptions of roles were collected and grouped into the twelve groupings presented below. Then four years of INCOSE symposium proceedings were scanned to ensure that most of the possible systems engineering roles were captured. The intent was to include roles applicable both to the typical DOD and aerospace environment and to less standard systems engineering environments such as smaller programs and commercial companies. Finally, the *Washington Post* newspaper's "High Tech" classified advertisement section was examined to determine what the world of employers considered systems engineering to be.

This paper is organized in four sections. First, the meaning of "systems engineering roles" is discussed. Next, twelve systems engineering roles are defined. These roles are then considered in relation to "life-cycle" and "program management" roles, the two major paradigms of systems engineering responsibility. Finally, the roles assumed in the first issue of *Systems Engineering* are characterized with respect to the twelve defined roles.

SYSTEMS ENGINEERING ROLES VERSUS SYSTEMS ENGINEERS' ROLES

There has been much discussion about whether INCOSE is about systems engineers or about systems engineering. The difference here is whether all engineers should be included as systems engineers or just those with a "systems engineering" title or job. If all engineers can do systems engineering, it is by considering the item they are engineering to be a system within a larger system, and by investigating operational issues, interfaces, and architecture of both systems.

The confusion comes about in part because many of the companies who have provided the basis for the field of systems engineering have done so by creating "systems engineering" departments or other groups, and charging them with systems engineering the product that is delivered to the customer. Within the company, those engineers who work on the subsystems or elements that are integrated to form the larger system are often not called or considered "systems engineers,"

[1] "What is the Value of Systems Engineering," at the Washington Metropolitan Area chapter, October 10, 1995. Panelists included Dorothy McKinney, Andrew Sage, Dale Langston, and John Snoderly.

Published in the Proceedings of the INCOSE Sixth Annual International Symposium (Boston, Massachusetts, USA).

but rather "transmitter engineers," "electrical engineers," or "software engineers." Conversely, those with the title of "systems engineers" work in these companies only on the larger system, not the subsystems or elements.

In this paper, the distinction is not really relevant. Although the author's experience is primarily in such organizations with functional departments labeled "systems engineering" departments, it is recognized that all engineers can and probably should adopt a systems engineering approach. Most of the following roles need to be addressed for each system at some time.

SYSTEMS ENGINEERING ROLES DEFINED

Role Definitions. Table 1 shows the twelve roles described in this section. In the paragraphs below, each of these twelve roles is given a "short name" and abbreviation for convenience. This is followed by related alternative role names, separated by slashes. A more detailed description of the role follows.

Table 1. Systems Engineering Roles

Role	Abbr.	Short Name
1	RO	Requirements Owner
2	SD	System Designer
3	SA	System Analyst
4	VV	Validation/Verification Engr.
5	LO	Logistics/Ops Engineer
6	G	Glue Among Subsystems
7	CI	Customer Interface
8	TM	Technical Manager
9	IM	Information Manager
10	PE	Process Engineer
11	CO	Coordinator
12	CA	Classified Ads SE

1. Requirements Owner (RO) Role. Requirements Owner / requirements manager, allocater, and maintainer / specifications writer or owner / developer of functional architecture / developer of system and subsystem requirements from customer needs.

This role groups several requirements-related tasks together. The first is translating customer needs into specific, well-written requirements to which systems and subsystems (subelements, pieces, software and hardware, control items, etc.) can be architected and designed. Requirements duties include understanding all external interfaces and ensuring the functional architecture correctly captures the need.

As potential changes are generated, requirements owners assess impacts to the overall system and to the subsystems, based on which specific requirements must be modified. On military-standard-compliant programs, another task is creation and maintenance of subsystem specifications.

The formality of the requirements process may vary significantly with project size, degree of customer-imposed formality, and company culture. Larger projects depend on communications among more people and usually develop more formal processes in response.

2. System Designer (SD) Role. System Designer / owner of "system" product / chief engineer / system architect / developer of design architecture / specialty engineer (some, such as human-computer interface designers) / "keepers of the holy vision" [Boehm 94].

An engineer in this role creates the high-level system architecture and design and selects major components. Possible ways of building the system from pieces are investigated and compared to the system requirements, the system design is selected and fine tuned, needs for the next lower subsystems are described in detail, and it is confirmed that subsystems that can meet the specifications are available or can be developed. Because of the complexity of projects employing systems engineers, the emphasis tends to be on architecture, high-level design, integration, and verification, rather than on low-level development.[2]

This role comes into play conceptually after the requirements and functional architecture are developed by the RO (Requirements Owner). Of course, in reality the two tasks overlap, and the two roles interact, especially in the selection of subsystems and elaboration of subsystem requirements.

3. System Analyst (SA) Role. System Analyst / performance modeler / keeper of technical budgets / system modeler and simulator / risk modeler / specialty engineer (some, such as electromagnetic compatibility analysts).

System analysts confirm that the designed system will meet requirements. Typical analyses include system weight, power, throughput, and output predictions for hardware systems, and memory usage, interface traffic, and response times for software systems. Usually the more complex parts of the system need to be modeled in order to demonstrate that they will work properly and interface properly with the external world. Modeling also helps the systems

[2] Dr. Eberhardt Rechtin of the University of Southern California (USC) has commented that his "systems architect" role is distinct from the role of systems engineering. At USC, there is general agreement that systems architects have the roles shown in the "Rechtin" line of Table 2 and systems engineers have the roles in the "Friedman" line. George Friedman is also associated with USC.

engineer and others understand how the system will be operated.

Most systems engineers do some modeling, with the amount increasing with the availability of cheap and powerful simulation tools. The extent of analysis varies with the type of program: the more complex and riskier programs need more analysis.

4. Validation and Verification (VV) Role. Validation and Verification engineer / test engineer / test planner / owner of system test program / system selloff engineer.

VV engineers plan and implement the system verification program to ensure the system, as designed and built, will meet the specified requirements. In some organizations, systems engineers also write the detailed system test plans and test procedures. During the system verification process, questions usually arise as to what was supposed to have happened during a scenario. VV engineers are responsible for answering these questions in real time and, to the extent possible, for predicting such behavior in advance. VV engineers also are required to respond to anomalies with the best possible understanding of the system design. They must also know which experts to call when needed.

In some organizations, parts of the VV roles are performed instead by a separate System Test group, and, in other organizations, both a systems engineering group and a system test group have defined roles in the system validation program.

5. Logistics and Operations (LO) Role. Logistics, Operations, maintenance, and disposal engineer / developer of users' manuals and operator training materials.

This role captures the back end of the "cradle-to-grave" or "lust-to-dust" system life cycle. During the operational phase, systems engineers sometimes operate the system for the customer; more often, they serve "on call" to answer questions and resolve anomalies.

In addition to owning primary responsibility in the later phases of programs, LO engineers are usually expected to bring maintenance, operation, logistics, and disposal concerns to the requirements, design, and development phases. As creators of users' manuals, they need to understand most design aspects and all operational aspects of the system, and determine what users do and do not need to know about the system.

6. Glue (G) Role. Owner of "Glue" among subsystems / system integrator / owner of internal interfaces / seeker of issues that fall "in the cracks" / risk identifier / "technical conscience of the program" [Fisher 92].

In this role, the systems engineer serves as a proactive troubleshooter, looking for problems and arranging to prevent them. Since many problems happen at interfaces, this role involves a very close scrutiny of interfaces, particularly internal, subsystem-to-subsystem interfaces. While the designers of the subsystems struggle to make their subsystems do what they are supposed to, the G systems engineer is watching to ensure that each subsystem is not going to interfere with the others.

In hardware systems, negative effects such as electromagnetic incompatibility and passive intermodulation products can be caused by designers of structural subsystems inadvertently employing materials that adversely affect the system electronics. Incorrect software module interfaces can result in out-of-bounds conditions, race conditions, or inappropriate failure recovery sequences. In the G role, systems engineers need wide experience, meaningful domain knowledge, and an interest in continuous learning to stay a step ahead of problems like these.

7. Customer Interface (CI) Role. Customer Interface / customer advocate / customer surrogate / customer contact.

In a pivotal work in the field of systems engineering, [Rechtin 91] defines the Systems Architect role as a combination of the SD (System Designer) role and this role.[3] Systems engineers can be asked to represent the point of view of the customer, and to see that it is properly respected throughout the program. They can also serve as the interface to customer technical personnel in this role, striving to ensure the "right" system is built, and that the details are as customer-friendly as possible.

The CI (Customer Interface) role includes only the role of the engineers building a customer-deliverable product, not the full marketing process of a business or organization.

The CI role is handled quite differently by different organizations and businesses. Some prohibit engineers from talking to the customer, either because they believe that a customer would want to talk to someone at a higher level, i.e., a manager, or because they fear that technical people will "give it all away for free," promising a more thorough interpretation of contract requirements without negotiating additional funding. These organizations do not consider CI to be a systems engineering role.

[3] It should be noted that Dr. Rechtin considers the systems architect to be distinct from systems engineers. The systems architect is a neutral third party who can negotiate with both the customer and the builder. Many INCOSE systems engineers seem to take on the role of systems architect at some time, however, so it is included here as part of systems engineering.

Other organizations consider program systems engineers to be the primary interface for technical customers throughout the program. These systems engineers are assumed to keep customer technical personnel up to date on design decisions, rationales, issues, and concerns. They also tend to work the technical side of change requests, finding out what the customer intends and why, and communicating the impacts that are discovered. This is the bulk of the CI role.

Still other organizations rotate people trained as program systems engineers into a specific role in the new business development area. These systems engineers frequently write proposals, interacting with the builders of subsystems to assess feasibility and cost of various options. This is the CI role as expressed in the early life-cycle phases. Clearly these engineers need to work closely with those playing the SD (System Designer) role, who are also very active at this time. These are the "trappers" of the "trappers vs. skinners" model; they become "skinners" if they continue to work the program after contract award.

8. Technical Manager (TM) Role. Technical Manager / planner, scheduler, and tracker of technical tasks / owner of risk management plan / product manager / product engineer.

Technical management is one part of program management, which also includes controlling cost, scheduling resources, and maintaining support groups such as configuration management, computer network staff, and finance. The technical management part is sometimes assigned to a program systems engineering manager or to engineers responsible for the customer-deliverable system.

As the reach of systems engineering extends to commercial companies, a type of systems engineer called the Product Manager or Product Engineer appears. This role is similar to a Systems Engineering Manager role, with authority over a much smaller group of engineers, maybe only one. On a small project, the Product Manager or Product Engineer may wear more of a marketing hat and more of a cost and schedule hat than technical managers on large programs wear.

9. Information Manager (IM) Role. Information Manager (including configuration management, data management, and metrics).

Historically, some authorities have considered configuration management to be a systems engineering role. These are generally the authorities who lean toward the "program management" view of the systems engineering task. As information systems become more complex and more pervasive, it becomes more important for someone to view the overall information needs of the system, and even of the business. Thus, this role may grow to include data management and process asset management.

Organizations attempting to improve their capability maturity begin to define and capture metrics. The organizational set of program metrics can be considered information to be managed, preferably by someone with an enterprise-wide viewpoint.

10. Process Engineer (PE) Role. Process engineer / business process reengineer / business analyst / owner of the systems engineering process.

This is a fairly recent systems engineering role. Those who do systems engineering are also expected to document, follow, own, and improve the project's and the organization's systems engineering processes. This role also calls for defining and capturing systems engineering metrics.

Recently the "reengineering" of industry has called for a cadre of "reengineers" to be developed, and those trained in systems engineering have sometimes been asked to participate, because the skills of designing a complex product can be applied to designing business processes as well.

11. Coordinator (CO) Role. Coordinator of the disciplines / tiger team head / head of integrated product teams (IPTs) / system issue resolver.

Because systems engineers have a broad viewpoint, they are sometimes asked to coordinate groups and resolve system issues, at least to the point of seeking consensus, or making recommendations, when consensus cannot be achieved among the participants. Even if there are no "systems engineers," coordination can be considered vital to the engineering of a complete system. This role may be permanent, defined in terms of team or discipline coordination, or transitory, established to solve a specific problem and then dissolved. Team leadership and the ability to facilitate groups in developing their own leadership skills and working norms are skills that are more necessary for this role than for others.

12. "Classified Ads Systems Engineering" (CA) Role.

This role was added to the first eleven in response to frustration encountered when scanning the classified ads, looking for the INCOSE-type of systems engineering jobs. Approximately half of the advertisements for "systems engineers" in a recent newspaper seemed to be asking for other things. For example, note the words in selected ads:

> "Skills must include shell scripting, SQL, performance analysis, and network integration."

> "...five years of solid analytical & debugging expertise in a telecommunications environment"
>
> "To analyze and develop systems level software in C/C++ and UNIX scripts."
>
> "Object-Oriented/Design/Analysis/Programming... RDBMS (Oracle), ...CICS/PLI, ...STAIRS/ Search Manager..."
>
> "Provide UNIX Administration and service delivery for our ... Internet service"
>
> "Provide design, implementation, and ongoing support for Managed and Non-Managed Private X.25, Frame Relay, and ATM Networks..."

So this role represents "other," compared to the first eleven systems engineering roles. It is heavily weighted toward computer systems and may have developed from the need for programmers to adopt broader viewpoints, first as software engineers and later looking at whole computer systems.

In some cases, this role may be more of an illusion, when the companies are simply asking for people trained in the first eleven roles, but who also have experience working in the computer domains listed. Another possible explanation is that the term "systems engineer" meant, at one time, the person responsible for maintaining the mainframe computing system for an organization, so some advertisers may be looking for a similar skill, with updated applications such as Internet and client-server architecture. Still other advertisers may be seeking someone less "researchy" and more "applied" than a "computer scientist," and yet less narrowly focused than a programmer. Since there is no generally accepted name for this type of person, the name "systems engineer" may be attached to this role.

It is likely that all three of these explanations (and others not yet considered) represent the truth about this role. Although the details are unresolved, it is useful to retain this role as a reminder that there are more definitions of systems engineering than the first eleven roles.

ROLE COMBINATIONS

Life-Cycle Versus Program Management Roles. Early INCOSE symposia discussed two views of the role of systems engineering. One view considers the job of systems engineers to be coordinating, tracking, and managing the engineering of the system and its subsystems. The other view considers the job to be a set of specific life-cycle tasks. The arguments continue, and official INCOSE definitions of "system" and "systems engineering" did not appear until mid-1995 and early 1996, respectively.

This paper attempts to describe the current understanding as to what systems engineering is in terms of individual roles. The "common language" sought by [Brill 94] can be facilitated by showing which of the twelve roles are life-cycle roles, which are parts of the program management view, and which belong in both.

Life-Cycle Roles. As INCOSE has attempted to define a standard systems engineering process, specific tasks have been defined that need to take place during the system life cycle. The roles that describe these tasks include RO (Requirements Owner), SD (System Designer), VV (Validation and Verification engineer), LO (Logistics and Operations engineer), and SA (System Analyst). Here, the first four take place approximately in that order, and the SA role is considered to take place either at defined times in the life cycle or continuously throughout.

Of course, life-cycle roles no more follow a strictly-interpreted waterfall model than the systems engineering process does. Rather, the emphasis on the roles and the resources devoted to them evolves in time.

Program Management Roles. [DSMC 90] describes systems engineering as primarily program management. Roles included are TM (Technical Manager), G (Glue), IM (Information Manager), CO (Coordinator), and possibly CI (Customer Interface). In addition, some of the life-cycle roles above might be included, though as less important roles relegated to lower-level engineers.

This definition of systems engineering would arise naturally from the fact that only complex (which usually means *large*) systems projects tend to have a separate systems engineering group, and from the fact that the technical tying-together of the system is difficult to separate in practice from the business tying-together of groups building the system components.

When a group of systems engineers performs technical management tasks, it tends to function as the technical arm of the program manager. One engineer seasoned in this type of systems engineering described his role as "making life difficult for the program manager when the right technical decision costs more." Technical advocacy is expected as part of the G (Glue) role but, of course, any adversarial relationship between program management and systems engineering would be detrimental to program progress and the achievement of business objectives. Both groups want the system to work well and be delivered on time, at a reasonable profit to the business. On a small program, both these roles might be fulfilled by the same person.

ROLE ALLOCATION

Systems engineering is a naturally broad field. No one person will perform all twelve roles at once, and many engineers will never perform all the roles even over the course of an entire career. Different organizations disagree as to whether some of the roles are appropriate for a group designated as "systems engineers," but most will consider some of them to be vital to their concept of systems engineering. It is hoped that these definitions will provide a common language about roles, which may be helpful in discussions about the nature of systems engineering.

When systems engineering is performed on a large program, parts of the entire systems engineering task are usually allocated to individual engineers. Each person identified as a systems engineer might, for example, be assigned a subsystem or a software component to oversee. In addition, each might be asked to coordinate the development of something that crosses subsystem boundaries, such as an operational concept document, a test plan, a risk identification document, or a performance budget.

In a similar manner, the twelve roles are usually allocated among people or groups. For example, a performance analysis group might be established that "owns" specific analyses from the set listed under the System Analyst role. Or one or a small number of senior systems engineers might be assigned to serve as the systems architects, included here as part of the System Designer role.

In addition, because priorities vary from project to project, resources for accomplishing the roles will vary. Systems based on legacy systems will have less architectural flexibility and less SD (System Designer) effort. High risk systems will require more systems analysis (SA). Programs with very involved, technical customers will need to devote more resources to CI (Customer Interface). The interactions among the roles mentioned below will also need to be taken into account when planning the systems engineering effort.

ROLE INTERACTIONS

Dividing any complex body of knowledge into pieces is something of an art, and specifying systems engineering roles is no exception. Although the boundaries between roles have been chosen for cohesion and low communication, significant interactions still occur. Some of the more noteworthy are as follows:

Risk. Risk *analysis* has been included here as part of the SA (System Analyst) role, risk *identification* as part of the G (Glue) role, and the *management* of risk (workarounds, mitigation plans) has been included in the TM (Technical Manager) role.

Design Reviews. As part of the TM role, systems engineers set up the reviews and track action items. For externally-required design reviews, the CI (Customer Interface) role ensures appropriate customer attention, and the G (Glue) role is involved to ensure that the design is thoroughly reviewed by appropriate parties.

Metrics. The PE (Process Engineer) role identifies needed process metrics, establishes the measurements that need to be taken, and defines the metrics algorithms. Systems engineers performing life-cycle tasks take the measurements, and technical managers use the metrics for decision making.

Customer Interaction. The LO (Logistics and Operations) role is very customer-focused, attempting to ensure usability and operational suitability. ROs (Requirements Owners) need to ensure the specified, designed, and built system will be what the customer wanted, and the CI (Customer Interface) role encompasses the various interfacing that should be done with the customer throughout the program.

THE ROLES IN INCOSE PAPERS

Table 2 compares the roles above to assumptions evident in the inaugural issue of *Systems Engineering*, the Journal of INCOSE. Some of the journal papers addressed systems engineering issues orthogonal to roles, but a few role assumptions could be deduced. Because some roles lack significant support in journal papers, other sources that address these roles more directly are also included in the table.

Table 2 shows that the first eleven roles are all considered to be systems engineering roles by some denizens of the INCOSE community, but no one included them all. The "semantic jungle" of [Brill 94] has clear causes, given that no two authors have the same definition of what roles systems engineers have. The most well-covered roles are the ownership of requirements (RO), the design of the system itself (SD), and systems analysis (SA), with VV (Validation and Verification) and G (Glue) close behind. These, which are nearly the same as the life-cycle roles mentioned above, are probably the most "basic" systems engineering roles, but Table 2 shows there is disagreement about even them.

FUTURE EFFORT

In addition to resolving underlaps and overlaps among the roles, more work should be done to understand how the commercial arena utilizes systems engineering. Biases in this paper may stem from the author's history working on development programs—differences in the maintenance and COTS integration environments should be investigated.

Table 2. Systems Engineering Roles Apparently Assumed in Various INCOSE Papers

Role Reference	1 RO	2 SD	3 SA	4 VV	5 LO	6 G	7 CI	8 TM	9 IM	10 PE	11 CO
Bahill 94		σ	✓								
Beam 94	✓	✓	✓	✓	✓	✓					
Blanchard 94	✓	✓		✓	✓	✓					✓
Boehm 94		σ	σ			σ					
Dick 94	✓		σ				✓			✓	
Fabrycky 94	✓	✓	σ	σ							
Friedman 94	✓		✓	✓	✓	✓		✓	✓	✓	
Grady 94	✓	✓	✓			σ	✓	✓		✓	
Hatley 94	σ	σ	σ								
Lacy 94	✓							σ		σ	
Lake 94	σ	✓	✓	σ	✓	✓	✓	✓	✓		✓
Mar 94	σ	σ	✓	σ				✓	σ		
Rechtin 94		σ	✓			✓	σ				
Sage 94	✓	✓	✓					✓	✓	σ	
Wymore 94	✓	✓	✓	✓	✓						
Bate 95 (SE-CMM)	σ	σ	σ	σ		σ	✓	σ	✓	✓	σ
CAWG 95 (SECAM)	σ	σ	σ	σ		✓		σ	✓	σ	σ
DSMC 90	σ		✓	σ	σ	σ		σ	✓		✓
Matty 95	σ	σ					✓	σ			σ
McKinney 95	σ	✓	σ			σ			✓		✓
Sheard 95	σ	✓		σ		σ	σ	✓			σ

σ=Primary Assumption, ✓=Secondary Assumption

The paper, “The Value of Twelve Systems Engineering Roles” [Sheard 96], a companion paper in this Proceedings, discusses the value of systems engineering from the point of view of the twelve roles. Additional roles that have been identified during discussions of value, such as “Systems Engineering Educator,” and “Systems Engineering Evangelist,” should be considered for inclusion, and their value defined.

CONCLUSIONS

Because assumptions of the papers in the inaugural issue of the INCOSE journal can be characterized by these roles, the twelve roles listed here are likely to be a good starting point for a definition of what systems engineering is and does. Organizations can determine whether their definition of systems engineering has more of a life-cycle or a program management outlook.

An individual working in, or manager organizing, a systems engineering effort should be prepared to answer the following questions:

- Which systems engineering roles are vital and which are of secondary importance?
- What are the roles of systems engineering considered to be in my organization, and should we consider other roles as well?
- Can an awareness of roles help my organization improve its systems engineering capability?

REFERENCES

Bate, Roger, et. al., *A Systems Engineering Capability Maturity Model*, Version 1.0, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1995.

Capability Assessment Working Group (CAWG). “Systems Engineering Capability Assessment Model,” Version 1.40, *Proceedings of NCOSE*, vol. 2, 1995

Defense Systems Management College, *Systems Engineering Management Guide*. US Government Printing Office, January 1990.

Fisher, Jack. “Systems Engineering for Commercial Space Programs,” *Proceedings of NCOSE*, 1992.

Mar, Brian, “Back to Basics,” *Proceedings of NCOSE*, 1992.

Matty, George E., Jr. “The Benefits of the Systems Engineering Process in Concurrent Engineering,” *Proceedings of NCOSE*, 1995.

McKinney, Dorothy. “The Value That Systems Engineering Can And Should Provide,” *INSIGHT*, INCOSE, Dec. 1995.

Rechtin, Eberhardt. *Systems Architecting, Creating and Building Complex Systems*, Prentice-Hall, Inc., New Jersey, 1991

Sheard, Sarah A. Team Structures for Systems Engineering in an IPT Environment, *Proceedings of NCOSE*, 1995.

Sheard, Sarah A. The Value of Twelve Systems Engineering Roles, *Proceedings of INCOSE*, 1996.

The following papers are all from *Systems Engineering*, Vol. 1, No. 1, 1994:

Bahill, A. Terry and Chapman, William L. “Understanding Systems Engineering Through Case Studies.”

Beam, Walter R. “Evolutionary Systems: Arguments for Altered Processes and Practices.”

Blanchard, Benjamin S. “The System Engineering Process: An Application for the Identification of Resource Requirements.”

Boehm, Barry. “Integrating Software Engineering and Systems Engineering.”

Brill, James H. “Systems Engineering—A Semantics Jungle.”

Dick, Michael J. “System Thinking: Breaking Murphy's Law.”

Fabrycky, Wolter J. “Simulation and Direct Experimentation in System Design Evaluation.”

Friedman, George J. “Systems Engineering's Crucial Juncture.”

Grady, Jeffrey O. “System Engineering Dichotomies.”

Hatley, Derek J. “Current System Development Practices Using The Hatley/Pirbhai Methods.”

Lacy, James A. “Visualizing Systems Engineering.”

Lake, Jerome G. “Axioms for Systems Engineering.”

Mar, Brian W. “Systems Engineering Basics.”

Rechtin, Eberhardt. “Foundations of Systems Architecting.”

Sage, Andrew P. “The Many Faces of Systems Engineering.”

Wymore, A. Wayne. “Model Based Systems Engineering.”

BIOGRAPHY

Sarah A. Sheard has sixteen years’ experience in systems engineering. Ms. Sheard worked as a satellite engineer at Hughes Aircraft Space and Communications Group, and in software systems at Loral Federal Systems. Currently she manages the systems engineering effort at the Software Productivity Consortium in Herndon, Virginia, where she also consults and teaches in systems engineering, process improvement, and teams. Ms. Sheard received her MS in chemistry from the California Institute of Technology in 1979.