Wiley Series on Systems Engineering and Analysis
HAROLD CHESTNUT, Editor

Chestnut
Systems Engineering Tools

Wilson and Wilson
Information, Computers, and System Design

Hahn and Shapiro
Statistical Models in Engineering

Chestnut
Systems Engineering Methods

Rudwick
Systems Analysis for Effective Planning: Principles and Cases

Sunde
Communication Systems Engineering Theory

Wilson & Wilson
From Idea to Working Model

Reitman
Computer Simulation Applications

Rodgers
Introduction to Systems Safety Engineering

Miles
Systems Concepts: Lectures on Contemporary Approaches to Systems

---

Systems Concepts

LECTURES ON CONTEMPORARY APPROACHES TO SYSTEMS

1971 Lecture Series at the California Institute of Technology, Sponsored by the Divisions of Biology, Chemistry and Chemical Engineering, Engineering and Applied Science, and the Humanities and Social Sciences, the Faculty Committee on Relations with Industry, and the Industrial Relations Center

EDITED BY
Ralph F. Miles, Jr.
MEMBER OF THE TECHNICAL STAFF
JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY

A WILEY-INTERSCIENCE PUBLICATION

John Wiley & Sons, New York • London • Sydney • Toronto

1

Introduction

RALPH F. MILES, JR.

Member of the Technical Staff
Jet Propulsion Laboratory
California Institute of Technology

Ralph F. Miles, Jr., received his Ph.D. in Physics from the California Institute of Technology in 1963. At the Jet Propulsion Laboratory he was the Spacecraft System Engineer for the Mariner Mars 1969 Project. During the 1969–1970 academic year he was a Visiting Fellow in the Department of Engineering-Economic Systems at Stanford University. During the 1970–1971 academic year he was a Visiting Assistant Professor in Aeronautics and Environmental Engineering Science at the California Institute of Technology. Presently he is the Mission Analysis and Engineering Manager for the Mariner Jupiter/Saturn 1977 Project at the Jet Propulsion Laboratory.

Systems Concepts

Most people have intuitive ideas about the systems approach, or "systems engineering" as it is called in the more technically oriented contexts. Civil engineers have been constructing large systems for a long time—systems such as cities, roads, aqueducts, and pyramids. Today aeronautical, chemical, and electrical engineers design large technically complex systems with complicated man-machine interfaces. Computer programmers, biologists, economists, and sociologists all use systems concepts.

To a large extent these intuitive notions of systems are correct. After all, is systems engineering not just "good engineering," what we have been trying to do all along?

Systems engineering *is* good engineering. And beyond that it is more a change in emphasis than a change in content—more emphasis on defining goals and relating system performance to these goals, more emphasis on decision criteria, on developing alternatives, on modeling systems for analysis, and on controlling implementation and operation.

Collectively, systems concepts constitute a viewpoint and an approach involving the optimization of an overall system as distinct from the piecemeal suboptimization of its elements. In addition, the general class of systems concepts also includes a number of techniques, both methodological and analytical, which are involved in the design and operation of systems.

Webster's unabridged dictionary devotes one full column to the word "system" and its grammatical forms and synonyms. Some of the definitions relevant to our purposes are the following:

- A complex unity formed of many often diverse parts subject to a common plan or serving a common purpose.
- An aggregation or assemblage of objects joined in regular interaction or interdependence; a set of units combined by nature or art to form an integral, organic, or organized whole; an orderly working totality.
- A group of devices or artifical objects forming a network or used for a common purpose.
- An organized or established procedure or method or the set of materials or appliances used to carry it out.
- An organization or network for the collection and distribution of information.

For the purposes of this book, a system is defined as a set of concepts and/or elements used to satisfy a need or requirement. The idea of a system arises when one can associate a need with a capability for satisfying that need. Thus there are many kinds of systems: aerospace systems, sewer systems, administrative systems, cardiovascular systems, systems for gambling, and even systems for beating the system. As shown in Fig. 1, "systems engineering" is defined as the set of concepts and techniques which are necessary to proceed from the original system concept to the creation of the system or, more completely, to the satisfaction of the original need. Thus one can speak of the systems engineering of a planetary mission to Mars or the systems engineering of a more purposeful and efficient judicial system.

There is no clear-cut distinction as to the types of systems for which systems concepts are appropriate. In the spirit of Robert Machol and Ralph Miles in Chapter 3, the class of systems for which systems concepts are relevant have the following properties:

- The system is man-made.
- The system has integrity—all components are contributing to a common purpose, the production of a set of optimum outputs from the given inputs.

- The system is large—in number of different parts, in replication of identical parts, perhaps in functions performed, and certainly in cost.
- The system is complex, which means that a change in one variable will affect many other variables in the system, rarely in a linear fashion.
- The system is semiautomatic with a man-machine interface, which means that machines always perform some of the functions of the system and human beings always perform other functions.
- Some of the system inputs are random, which leads to an inability to predict the exact performance of the system at any instant.

A system need not exhibit all these characteristics in order for systems concepts to apply. Nevertheless, characteristics of this nature determine the degree of relevance and necessity for systems concepts.

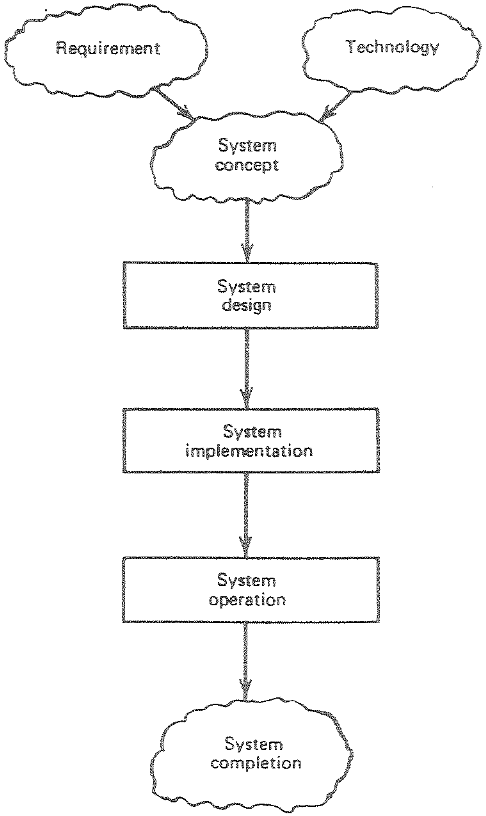


Figure 1 Systems engineering stages.

Figure 2 shows a hierarchy of system types. The base level of the triangle consists of the codification of our knowledge into the basic sciences, wherein we mean to include the soft sciences as well as the hard sciences. The second level of the triangle consists of the technologies, the skills and techniques we have acquired for applying our knowledge. At the third level are the technical systems, which apply, integrate, and manage technologies in a collective effort to achieve a technical goal. Above the technical systems are the civil systems, more directed toward social welfare goals—transportation systems and medical-care delivery systems for example. At the apex of the triangle are the social systems, whose goals are social welfare—management systems, systems of law and justice.

What identifies the level of a particular system? Obvious criteria are related to the amount of technical hardware involved. A more basic and underlying rule says that as one ascends the triangle, the goals or purposes become more and more related to social welfare. There are no social welfare criteria with respect to the basic sciences. No one asks how an atom ought to be. Social welfare judgment does enter into technology. Should a nation use the scientific knowledge of atom-splitting to build a nuclear power station? An atom bomb?

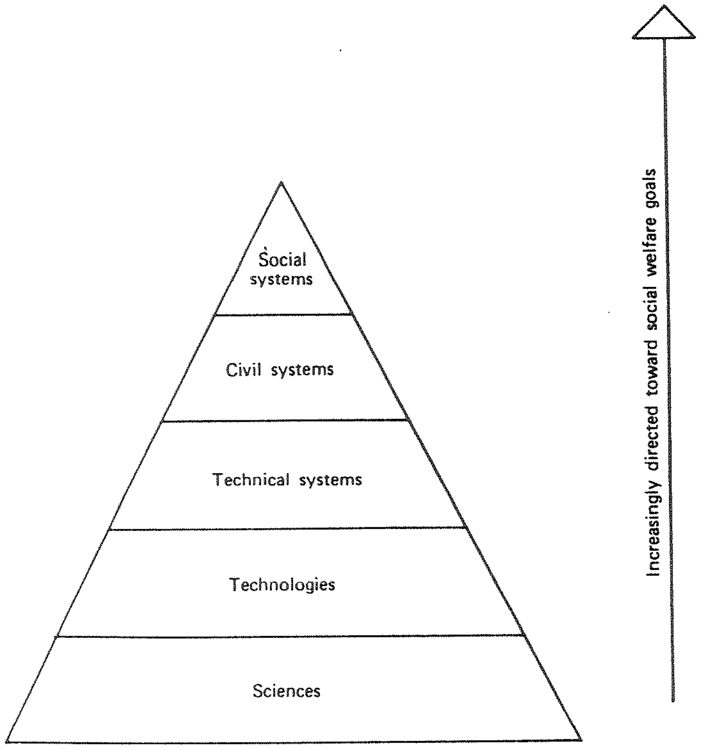


Figure 2 Hierarchy of systems, technologies, and sciences.

Purely technical systems are evaluated according to a defined and documented technical specification. The success of a space mission can be evaluated by comparing performance against technical objectives. Civil systems, although strongly pointed toward social needs, can be at least partially evaluated according to technical objectives. Transportation systems can be evaluated partially in terms of technical parameters such as cost-effectiveness and transit times, although the design of a civil system is fraught with social consideration. Whom shall the system serve? Who shall bear the costs? What levels of safety shall be required?

Social systems are primarily designed to enhance social welfare—governments instead of anarchy, law and order instead of lawlessness and chaos, education to remove ignorance. Easily measured social indicators often do not give a true picture of the effectiveness of a social system. Student-teacher ratios and cost of facilities do not in themselves measure the knowledge and skills imparted by an educational system.

Underlying every technology is at least one basic science, although the technology may be well developed long before the science emerges (e.g., glassmaking). Overlying every technical or civil system is a social system which provides purpose, goals, and decision criteria. Ultimately, of course, all systems involving people are embedded in a social system.

Solid-state physics is one of the basic sciences underlying the technology of transistor making. A machine to automatically make 10,000 transistors a day, or a factory to produce 10,000 transistor radios a day are examples of technical systems. A network of radio stations in an underdeveloped country, broadcasting educational material to illiterate peasants, would be a civil system. Whether an underdeveloped nation should allocate limited resources for such a system would be a question to be posed within the framework of the social system.

Overview of the Book

The organization of the book roughly breaks down into a set of chapters on theory and a set of chapters on application. There is not a clean break, however, primarily because all the authors have been involved in the practical application of systems concepts.

Chapter 2, by Simon Ramo, "The Systems Approach," provides an overview of systems concepts and their wide range of applicability. Systems engineering became accepted as a necessary activity in the design of complex systems around the time of World War II and has since been used in the design of many modern systems in industry, transportation, communications, and government. In Chapter 3, "The Engineering of Large-Scale Systems," Robert Machol and Ralph Miles discuss some of the guiding principles for systems engineering.

Designing and implementing systems requires that a large number of decisions be made—decisions concerning complex problems in complex environments. Decision analysis attempts to structure and analyze these complex decision problems in such a way so as to maintain the same logical relationships that would exist in an elementary problem. There is a long-standing question as to whether complicated real-world decision problems are amenable to modeling and analysis. It is undeniably true that the universe within which real-world decisions are made is richer in complexity than any model could ever hope to capture. Nevertheless, decisions *are* made and resources *are* allocated. The evidence is that all this occurs not on the basis of the complex universe but on the basis of relatively simple models of decision situations. What decision analysis purports to do is to incorporate into the analysis of real-world problems logical preciseness and the correct expression of preferences between alternatives. In Chapter 4, "Decision Analysis in Systems Engineering," Ronald Howard presents the decision analysis cycle and describes the roles of probability, utility, value, information, time, risk, and uncertainty in decision analysis.

Chapter 5, by Ward Edwards, "Divide and Conquer: How to Use Likelihood and Value Judgments in Decision Making," complements the preceding chapter on decision analysis in considering psychological concepts implicit in decision making. In any contrived, hypothetical decision situation, it can be shown that decisions arrived at through formal logic are superior to decisions made on the basis of intuition. Edwards develops the thesis that these same techniques of formal logic can be applied *in toto* to real-world decision situations and that this approach appears once again to be superior, though no proof is possible.

Large systems are inextricably complex. Existing techniques of analysis have been reshaped and new techniques developed to meet the needs for systems analysis. In Chapter 6, "Analysis Techniques for Operations Research," Philip Morse discusses some of the analysis

techniques and mathematical models useful to operations research and shows how they have been applied in a practical manner to systems problems.

The aerospace industry has made spectacular use of systems engineering—world-wide communication systems employing satellite links, Apollo systems that fly to the moon and return, spacecraft that travel hundreds of millions of miles to distant planets. The modern procedures for managing, planning, implementing, and operating complex aerospace systems were first developed on communication systems and ballistic missile systems. The procedures were extended to all military systems and were later incorporated by NASA.

The Jet Propulsion Laboratory has made a significant contribution to the development of systems engineering through its management of, and participation in, lunar and planetary projects. In recent years the Jet Propulsion Laboratory has applied its systems engineering expertise to civil systems problems such as medical-care delivery systems, environmental control, crime prevention, and the design of transportation systems. In Chapter 7, "Systems Engineering at the Jet Propulsion Laboratory," William Pickering presents the systems concepts that have been used in the design of lunar and planetary missions and discusses how these systems concepts have been transferred to the design of civil systems.

In Chapter 8, "Apollo: Looking Back," George Mueller discusses various problems encountered in the design and implementation of the Apollo program. He makes the point that in spite of the enormity of the Apollo program—the utilization of vast resources, the number of highly skilled people involved, the size and complexity of the systems—ultimately the resolution of the major problems rested with a very small number of people making extremely difficult decisions in the face of great uncertainties.

Planning-programming-budgeting as a management system originated in the Department of Defense during the 1960s. The essential aspects of this management system are: a careful specification and a systematic analysis of objectives; a search for the relevant alternatives, the different ways of achieving the objectives; an estimate of the total costs of each alternative; an estimate of the effectiveness of each alternative, of how it comes to satisfy the objectives; and a comparison and analysis of the alternatives. In Chapter 9, "Planning-Programming-Budgeting Systems," Henry Rowen discusses the success to date in applying this system in government.

In Chapter 10, "Systems Concepts in Social Systems," Robert Boguslaw presents the idea that social systems are designed by the components of the system. Man is the basic component of a social system, and the system exists to enhance his welfare. Requirements and constraints flow up the system hierarchy as well as down. Thus the social system designer is as much a negotiator and an arbitrator as he is a designer.

In the final chapter, "A Critique of the Systems Approach to Social Organizations," C. West Churchman reviews the history of organizational theory and the attempts to view these organizations as social systems to which systems concepts should be applicable. He concludes by stating a need for the application of the systems approach to systems analysis itself and by predicting that the systems approach of the future will incorporate what he describes as a dialectical learning process.

The Systems Approach

Quoting from Simon Ramo in Chapter 2:

> "The systems approach is a technique for the application of a scientific approach to complex problems. It concentrates on the analysis and design of the *whole,* as distinct from the components or the parts. It insists upon looking at a problem *in its entirety,* taking into account all the facets and all the variables, and relating the social to the technological aspects.

In applying the systems approach, the systems-oriented person recognizes that needs or problems originating at one level invariably have contributing factors at higher levels. Thus one should attempt to view the immediate needs or the exacerbating problem within a larger context. What are the factors that created the need or caused the problem to arise? Is the problem complete in itself or is it merely a manifestation of a larger, more fundamental concern? More police, supported by more exotic technology, could reduce the crime rate in city slums but would do little to get at the underlying economic and sociological problems. To minimize the occurrence of cancer, a medical doctor tells his patients not to smoke cigarettes. A behavioral scientist, in taking a systems approach, attempts to alleviate the psychological and sociological factors that led the patients to smoke.

To apply this approach to a systems problem of any consequence requires a vast wealth of knowledge and the interaction of a diverse number of talents. Thus the use of the much heralded "systems team" comprised of specialists from all the relevant technologies.

Using the systems approach and doing systems engineering involves solving a lot of problems, and for this reason it is valuable to examine

these systems concepts within a problem solving context. John Dewey stated the essence of problem solving some sixty years ago when he asked:

1. What is the problem?
2. What are the alternatives?
3. Which alternative is best?[1,2]

Every human being, be he the president of a multibillion dollar corporation or an aborigine of Western Australia, goes through life solving problems. Indeed, it can be said that life presents itself as a sequence of problems, terminating with one you can't solve! So there is absolutely nothing new about problem solving for human beings. What the systems approach purports to do is to logically structure the problem-solving methodology.

Dewey's formulation has today, largely through the influence of the communications and aerospace industries, evolved into the celebrated "systems approach." As shown in Fig. 3, a problem, need, requirement, or goal is quantified in terms of objectives that the system must satisfy and criteria that can be used to rank alternative systems. A process of system synthesis takes place in which a set of alternative systems are generated. Each of these systems is analyzed and evaluated in terms of the stated objectives and design criteria. The "best" or "optimum" system is then selected and implemented. Of course, in practice the process is extremely iterative, with results from later stages fed back to early stages to modify objectives, criteria, system options, and the like.

How would you use the systems approach? Let us assume that you have a situation which is concerned with a need or involves a problem. The systems approach asks you to do the following:

The Systems Approach

1. Goal definition or problem statement
2. Objectives and criteria development
3. Systems synthesis
4. Systems analysis
5. Systems selection
6. Systems implementation

Figure 3 The steps to the systems approach.

---

[1] John Dewey, *How We Think,* D. C. Heath, 1910.

[2] R. A. Johnson, F. E. Kast, and J. E. Rosenzweig, *The Theory and Management of Systems,* McGraw-Hill, 1967, p. 280.

1. Express your understanding of the situation in a logical, coherent manner: some words, a picture, mathematics if that is possible.
2. Develop a set of objectives and criteria that the system must satisfy in order to achieve the goal. If the situation involves a problem, state the characteristics that will exist when the problem has gone away.
3. Develop alternatives to resolve the situation; not just one, but a set of alternatives from which you can pick and choose.
4. Examine and analyze each of your alternatives with respect to your goals and criteria. Select the alternative which you prefer, and implement the solution.

The success of the systems approach up to this time indicates that the process works well when the system objectives can be clearly formulated, and when the required technologies and sciences are sufficiently mature. The objectives for the Apollo project can be clearly stated: "To place a man on the moon and return him safely before the end of the decade." During the 1960s four major system technologies required for the Apollo implementation reached maturity: large launch vehicles, vehicles for operation in space, a system for trajectory analysis and orbit determination, and a system for communications.

Two major factors inhibit the successful application of this design process to civil and social systems. The first and most fundamental is that system objectives and system criteria can rarely be clearly stated. How does one choose between two transportation systems: one which gives a fast, bumpy ride and one which gives a slow, smooth ride? By what criteria does one discern the optimality of an educational system?

The second inhibiting factor is the lack of maturity in the required technologies and sciences—the soft sciences. The "descriptive" man yet remains to be described. In addition to unknowns in physiological man, knowledge of psychological man is in a rudimentary state. The lack of this fundamental information results in one library, optimally designed for use, which has no windows and a second library, also purported to be optimally designed, which has many windows. Even less understood is the subject of what man ought to be—"normative man." Yet this question is implicit in the design criteria of every social system.

Systems engineering draws on all the concepts of the basic sciences and disciplines. The present state of the art for systems engineering is that there now exists a well-demonstrated methodology for integrating technical disciplines into technical systems. Civil and social systems now lie on the frontiers of systems engineering. It is possible that civil and social system designers may experience only limited success until the psychological and social nature of man is better understood.

It thus appears that the systems approach works well when certain conditions are satisfied. These conditions are met for the design of technical systems, but only partially so for the design of civil and social systems.

In some ways it is unfortunate, though almost inevitable, that the modern concept of the systems approach has evolved out of highly sophisticated technical programs. It has come into being burdened with the technical jargon of the aerospace business and cloaked with the mystique of computers and mathematics. Yet the true nature of the systems approach has the purity of simplicity and the believability of common sense.

The systems approach is just plain common sense in that each concept, each step, is the reasonable thing to do. The value of the systems approach is that it allows you to bring all these common-sense ideas together in concert to focus on the resolution of complex problems in complex environments.

The systems approach will not solve problems for you. Only you can do that. What the systems approach will do is permit you to undertake the resolution—your resolution—of a problem in a logical, rational manner. You are the one who must ascertain that a problem or a need exists. You are the one who must develop alternatives. You are the one who must develop the criteria for selecting a suitable alternative. The systems approach will not do any of these things for you.

Psychologists say that we human beings yearn for uniqueness, for the right to be an individual, a very special individual. The systems approach will give you this opportunity in a very rational framework. The systems approach will allow you to express your individualism rationally when you identify your problem, your alternatives, and your decision criteria.

logical advance and lagging social maturity. But in a decade or so, the public, the Congress, local governments, and leaders in industry and science may all be convinced of the value and importance of the systems approach. At about that time, we will face a new bottleneck: a shortage of good systems engineers, including, of course, the non-technologist members of the systems team—the economists, political scientists, psychologists, and sociologists. The work is difficult. Assembling technical and non-technical specialists into working groups with the wisdom and imagination required cannot go forward as rapidly as is desirable.

Still, it is pleasant to imagine a time when the only thing that retards the use of logic, objectivity, and all the tools of science is a lack of enough trained professionals. That will be the beginning of a golden age. Once most people are wedded to a logical and objective approach to social problems, the world will be a lot better, and science and technology can be used to the fullest on behalf of society.

BIBLIOGRAPHY

Ramo, Simon, *Cure for Chaos,* David McKay Co., Inc., 1969.

Ramo, Simon, *Century of Mismatch,* David McKay Co., Inc., 1970.

Ramo, Simon, "New Dimensions of Systems Engineering," *Science and Technology in the World of the Future,* A. B. Bronwell (Ed.), Wiley, 1970, pp. 127-146.

3

The Engineering of Large-Scale Systems

ROBERT E. MACHOL

Professor of Systems
Graduate School of Management
Northwestern University

RALPH F. MILES, JR.

Member of the Technical Staff
Jet Propulsion Laboratory
California Institute of Technology

Robert E. Machol is a Professor of Systems in the Graduate School of Management, Northwestern University. He is the editor of *System Engineering Handbook* and *Information and Decision Processes* and is a coauthor of *System Engineering,* an introduction to the design of large-scale systems. He was formerly the President of the Operations Research Society of America, the Head of the Department of Systems Engineering at the University of Illinois, and was the Vice President of Conductron Corporation. For Ralph Miles' biography, see p. 1.

Systems Definition

Many books on systems engineering have been written, and in each the authors have been faced with the problems of defining the term "system." There is no unanimity in these definitions, and it does not appear that any universally acceptable definition is likely to emerge in the near future. The most that can be asked is that the author delineate the area about which he is discoursing so that the reader can have an indication of their common areas of interest.

To pick one example, Johnson, Kast, and Rosenzweig define a system as follows: ". . . an array of components designed to accomplish a particular objective according to plan."[1] This very general defini-

---

[1] R. A. Johnson, F. E. Kast, and J. E. Rosenzweig, *The Theory and Management of Systems,* McGraw-Hill 1967, p.113.

This chapter is based on material in "Methodology of System Engineering" by R. E. Machol in *System Engineering Handbook,* McGraw-Hill, 1965, pp. 1-3 to 1-13. Portions are reprinted with permission of the publisher.

tion is, in fact, too general for our purposes. It encompasses entities which we will not regard as systems or for which our systems concepts have little relevance.

The following characteristics are typical of those that restrict the class of systems to those of interest.

1. The system is man-made, and it incorporates equipment, computer software, procedures, and the like. This eliminates anthills, river basins, universes, and many other interesting "systems."

2. The system has integrity—all components contribute to a common purpose, the production of a set of optimum outputs from the given inputs. What this purpose is, how we define optimum, and even the nature of the inputs will often be unknown at the start of the system design process, and their elucidation will be an important part of the task. Rigorously applied, this criterion would exclude cities and, indeed, nearly all social systems.

3. The system is large—in number of different parts, in replication of identical parts, perhaps in functions performed, and certainly in cost; such things as the ignition system in an automobile are thus excluded.

4. The system is complex, which is here taken to mean that a change in one variable will affect many other variables in the system, rarely in a linear fashion; in other words, the mathematical model of the system will be complicated. This eliminates systems which are merely large, such as a bridge or highway (apart from considerations of traffic flow).

5. The system is semiautomatic, which means that machines and computers perform some of the functions of the system, and human beings perform other functions of the system. The large, completely manual system (pyramid building with slaves or a large data bank maintained by clerks) is eliminated because it is too inefficient for our interests. The large, completely automatic system is eliminated because it either does not exist, or else exists only as a subsystem of a larger system with an essential man-machine interface.

6. The system inputs are probabilistic, which leads to an inability to predict the exact load or performance at any instant. In some cases the rate of input is predictable (e.g., in an automatic factory), but even here there are difficulties in design because of unpredictable variability in such things as environment and raw materials.

7. Many systems, especially the most complicated, all-encompassing systems, are competitive. In military systems a rational agent (the enemy) is trying to destroy or reduce the effectiveness of the system; in business systems ordinary competition, or in public service systems cheating or mere noncooperation, have similar effects.

It must be understood that no one of these characteristics is necessary and no subset is sufficient. A system, much like beauty, lies in the eye of the beholder. There is no generally accepted definition which will separate systems from nonsystems. A decade ago, the title of "system," within our context, was usually construed to apply only to technical systems, but today the term has been extended to civil and social systems.

An example of a system which would appear to be amenable to straightforward analysis might be a transportation system for intercontinental travel. One could imagine many alternative systems, ranging from ships to rockets, but let us assume that criteria such as minimum travel time, cost effectiveness, and technical feasibility dictate that the optimum alternative uses a large jet airplane of the Boeing 747 class.

Now, the system does not consist of just the 747 airplane; pilots are needed to fly the plane, stewardesses are needed to help the passengers, a ground crew must prepare the plane and load the baggage. In addition, the plane must take off and land on airfields. While in flight the plane occupies an airlane, hopefully to the exclusion of other planes. All this requires facilities, personnel, and procedures. Passengers do not appear as if by magic, in one-to-one correspondence with the number of seats on the plane. Thus there must be a reservation system, a subsystem of our transportation system, to insure that an adequate, but not too many, number of passengers will be present for the flight.

All these activities must be managed by an organization that operates and maintains the plane, and ensures that personnel, supplies, and facilities are available in the right number at the right time. The proper management of this operation requires that estimates of future requirements be made years in advance, based on a mass of data concerning past operations and on indicators of future trends.

Finally, there must be organizations, desireably few in number, concerned with the overall resources invested and the overall return obtained. At one level this involves the stockholders and the management of the airline. At a higher level it involves the national government, or a consortium of governments. It is they who must provide the overall objectives and decision criteria for the system and make the difficult trade-offs between public and private investment, convenience versus efficiency of operation, user versus nonuser considerations, and profit versus safety and reliability.

Thus what started off to be a straightforward technical system with seemingly well defined interfaces on closer inspection becomes a subsystem treaded through many larger systems, with system objectives so complex that we are left with limited prospects of realizing complete answers to even such basic question as, "Do the benefits equal the resources expended?"

Systems Design

Consider the design of a system. Someone hands you $100 million and asks you to make a system to control air traffic, or to connect 70 million telephones by direct dialing, or to perform some other function. What do you do next?

The designer of such a system immediately confronts a dilemma, because the problem of designing a large-scale system is overwhelming, if it is attacked all at once; yet, if the attack is piecemeal, it is unlikely to be successful. The hope of the designer is that the problem can be subdivided in such a way that the parts can be handled somewhat separately, and that ultimately these parts can be rejoined in a straightforward manner to form the total system. This subdivision can take place simultaneously in a number of conceptually different ways. In particular, there are:

1. The logical steps of systems design.
2. The chronological phases of systems design.
3. The functions of the system.
4. The components of the system.

In Chapter 1, Ralph Miles lists the steps of the systems design process as: (1) goal definition or problem statement, (2) objectives and criteria development, (3) alternative synthesis, (4) systems analysis, (5) systems selection, and (6) systems implementation. These are the logical steps of systems design, but rarely can they be performed in this order. Logically, one must formulate the problem before one solves it. In fact, one performs both functions simultaneously throughout the systems design process. Because the problem cannot be adequately formulated until it is well understood and cannot be well understood until it has been more or less solved, the two are inseparable. Thus the design of any system is extremely iterative, with the designer proceeding from problem formulation to solution to problem reformulation and so on, with each cycle producing a more refined, better understood, and, in principle, more optimal system.

The chronological phases of systems design can be ordered as definition, design, implementation, and operation. The definition phase involves an analysis of the requirements and selection criteria, a generation at the systems level of a range of feasible alternatives, and an analysis of the best alternatives. The conclusion of the definition phase comes with the selection of one systems alternative, and with a gross understanding of the implications of the selected alternative with respect to performance, cost, schedule, risk, required technology development, system lifetime, interfaces with other systems, and so forth.

The design phase starts with the product of the definition phase, a grossly defined system, and proceeds to define, design, and analyze the system down to the level such that all documentation exists for the complete creation of the system. The implementation phase brings the system into being. This phase includes the procurement of parts and materials, fabrication and assembly of hardware, coding and validation of computer software, and training of personnel. The implementation phase ends with the system level tests or review processes which are required to certify the system for operation. The operations phase starts with the first application of the system to its stated purpose and continues through to the final phase-out of the system at the end of the life-cycle.

While these phases logically follow one another, in practice, there will always be some overlap. The degree of overlap that is permitted in the chronological phases of a system life-cycle is derived from trade-off considerations between speed of system implementation and a desire for cost-effectiveness and risk-minimization. Large military-systems procurements have been carried out under both philosophies, and there are advocates and adversaries of both sides of the issue.

It is often convenient to subdivide a system along functional lines. NASA divides its deep space communication network into a set of six functional systems which cut across international, administrative, and facility boundaries. The six systems are: tracking, telemetry, command, simulation, monitor, and operations control. Hospitals divide their operations into functionally differing services such as surgery, internal medicine, pediatrics, and so on.

Perhaps the most tangible way of subdividing a system is by its respective components. Nevertheless, even here care must be exercised to achieve a breakdown which will aid and not detract from the designer's efforts. The natural and desired boundaries should lie at points that minimize the interaction across the interfaces. It would be sheer folly to subdivide the design efforts for a complicated electronics system by part type, for example, a resistor subsystem, a capacitor subsystem, and so on, even though such a subdivision might make the most sense for the parts procurement phase.

These four subdivisions—logical, chronological, functional, and component—often merge in the system design process, as demonstrated in the following sentence: "Determine the performance requirements (logical subdivision) for the design (chronological subdivision) of a transmitter (component subdivision) for the communications link (functional subdivision)."

Given that one does proceed in this manner—subdividing the problem in convenient and productive ways, then going through a process of synthesis and analysis, then recombining to form a new system which is more optimal than the results of the preceeding iteration—the question then is raised, "How does one stop?" Or does this iteration go on ad infinitum, with each cycle yielding one more increment of optimality? What one should realize, of course, is that the system under design must be viewed within the context of a larger system—a system which includes the resources being expended in the design process. Now the answer to the question becomes clear. The iterations of the design cycle cease when the marginal return of an iteration no longer exceeds the marginal cost of the iteration. Thus we conclude this section with the seemingly contradictory statement that no optimally designed system is optimal!

Principles of Systems Design

The fundamental principle of system design is simply to maximize the expected value. Obviously this requires considerable interpretation in any particular case, but at least the expected value has a succinct and well-understood definition. Where one has the choice of supplying too much of something (resulting in excessive cost) or too little (with the possibility of a penalty if it proves inadequate), this rule gives a guideline, and this kind of thing is done continually in the design process.

Thus we have "trade-off analyses," in which, for example, for an airplane we might compare increased takeoff power (for a potential increase in payload) versus decreased fuel economy (for a potential decrease in payload). In such a two-parameter analysis it is conceptually simple to find a maximum; in a complicated systems situation we would have to trade also with dozens of other parameters, with pairwise comparisons being totally inadequate. This leads to "cost/effectiveness studies," in which we attempt to maximize the effectiveness of the system (or its expected value) for fixed cost, or to

minimize the cost for fixed effectiveness. Because it is generally impossible to find a single number which realistically represents the effectiveness of a complex system, there is a good deal of subjectiveness, as represented by judgment, as well as objectiveness, as represented by analysis, in systems engineering.

The principle of suboptimization states that optimization of each subsystem independently will not lead in general to a system optimum, and that improvement of a particular subsystem actually may worsen the overall system. Since every system is merely a subsystem of some larger system, this principal presents a difficult, if not insoluble, problem,—one that is always present in any major systems design. We will discuss this point further in one of the following sections.

The principle of centralization refers to centralization of authority and decision making, that is, to centralization of information as distinguished from material. Most organizations are built on the principle that routine inputs are handled at a low echelon, with higher echelons being informed so they may veto specific decisions or change policy on general decisions, if it is appropriate (this is sometimes called "management by exception"). Nonroutine decisions are passed to higher echelons for decision. This decision hopefully establishes a policy so similar decisions in the future will become routine. A difficulty arises only when the speed required for making the decision exceeds the speed with which the information may be communicated to the higher echelon and the decision made there and transmitted back. Thus, as speeds of communication and decision-making increase, the disadvantages of centralization decrease. With the improvements in computers, communications, displays, and theories of decision-making, the optimum in the continuum between centralization and decentralization moves more in the direction of centralization in our complex systems.

The principle of events of low probability is related to the fact that no system can be all things to all people, all of the time. The principle states that the fundamental mission of a system should not be jeopardized, nor its fundamental objectives significantly compromised, in order to accommodate events of extremely low probability. Yet one frequently hears: "the most trivial detail may be the key to the entire intelligence picture; therefore, the system must be able to store and process every conceivable intelligence input" in spite of the fact that the resulting system is too complex to be workable. In other words "the soldier in the foxhole may have urgent requirements for the airborne reconnaissance information; therefore, the entire data processing system must be airborne, and provision supplied for air drop of finished data to the front lines" in spite of the fact that the resulting allocation of weight to airborne data processing equipment will seriously compromise the reconnaissance performance. The system engineer can sympathize with the soldier in the foxhole and the commander who is sensitive to his needs, but he should insist on reasonable compliance with the fundamental principle of maximizing the expected value of the system.

In many systems, a compromise is possible: the system can be designed to handle most events automatically, and to sound an alarm which calls for manual intervention when an uncommon event occurs which is beyond its capablities. For example, an automatic mail-sorting system would throw out, for manual sorting, those letters which were not of standard size, shape, or location of address. Such a system might handle 95 percent of the mail automatically, at a cost much less than that of 100 percent manual handling and enormously less than 100 percent automatic handling. Similarly, when you reach a wrong number through (automatic) direct-distance dialing, you simply call an operator for (manual) rectification of the error.

Models for Systems Engineering

A model, in principle, is a substitute for the real thing. Models are used as tools to gain knowledge through analysis and as a means of conveying information. A model may be used in lieu of the real thing for any of a number of reasons: economy—it may cost less to derive knowledge from the model, availability—the model may represent a system which does not yet exist or cannot be manipulated, information—the model may be a convenient way to collect or transmit information. Models form an important part of systems concepts because economy, availability, and information are all important factors in the design and analysis of large, complex, and dynamic systems.

There are many ways of classifying models which are useful for systems design and analysis. Three that we shall consider are simulation/symbolic, structural/empirical, and descriptive/normative.

Simulation models replicate a system in function or form. A drawing can be said to be a simulation of a system because it looks like the system. An analog computer simulates a system in that its parameters have the same time history as the system. A system test program is a simulation of the operations phase in which the functional and environmental interfaces of the system are simulated, and the system is tested in its operational modes.

Symbolic models have no physical or functional resemblance to the system. Symbolic models use ideas, concepts, and abstract symbols to represent a system, as expressed in the form of words, graphs, or mathematics. The documentation for a system would represent a symbolic model, as would the mathematical representation of the system operation.

Models may be both simulation and symbolic, depending on the point of view. Computer programs of complicated systems, which are obviously symbolic models, are often called simulation models because they simulate many parameters of the system and, in some cases, may actually duplicate the software portions of the system.

Another classification of systems models describes them as structural or empirical—structural, if the parameters and functional relationships of the model have direct correspondences with the elements of the system; and empirical, if the parameters of the model are adjusted to give the model correspondence to the system, but the parameters in themselves bear no relationship to the elements of the system. A Taylor series expansion of system response or a linear regression analysis of data would be examples of empirical models.

An important classification is the differentiation between descriptive and normative models. A descriptive model describes a system without making any assessment of the system's value or of the system's performance. Normative models describe a system as it would be if it satisfied some criterion of optimality. The design requirements or contract specifications for a system capability would represent a normative model of the system. The degree to which the descriptive model of a system corresponds to the normative model would be a measure of the optimality of the described system. Normative models provide goals for systems design and systems operation, whether or not, in fact, they are achieveable.

A simple mathematical model of a system can be represented by a transformation between an input set and an output set: $y = S(u)$ (Fig. 1).

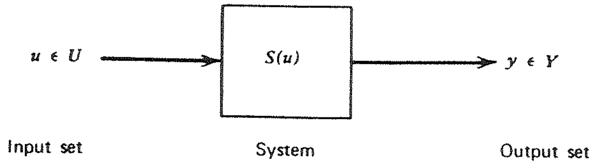


Figure 1 A mathematical model of a system.

This simple model will suffice to demonstrate an important concept in mathematical modeling, the division of math models into linear and nonlinear systems.

A system model is linear if and only if the input and output sets of the system can be represented as elements of linear vector spaces and the principle of superposition holds for the system relationships between the input and output sets, that is, the output of the system for a set of combined inputs is equal to the sum of the outputs for the individual inputs. More precisely, for a linear system with two inputs u_1 and u_2:

$$y = \alpha y_1 + \beta y_2$$

where

$$y_1 = S(u_1)$$

$$y_2 = S(u_2)$$

$$y = S(u) = S(\alpha u_1 + \beta u_2)$$

and α, β are scalars. In graphical form, this is shown in Fig. 2.

The mathematical techniques for linear models form an important part of systems analysis for a number of reasons. Many systems are linear, and many nonlinear systems exhibit a linear response over a restricted range of the system parameters. Linear systems theory is most general in application, while nonlinear systems analysis must often be applied on an *ad hoc* basis, with results which cannot be generalized to a broad class of systems. Finally, linear systems theory is conceptually easier to understand, and in fact forms a starting point for many nonlinear techniques.

If the internal nature of a system varies as a function of time or of prior inputs, then the nature of the system is called the *state* of the system. The state of an electrical circuit can be characterized by specifying the amount of energy in the circuit, that is, charge on capacitors and current through inductors. The state of an economy might be characterized by the available workforce and functioning facilities and businesses. The output of such an economy might be characterized by gross national product.

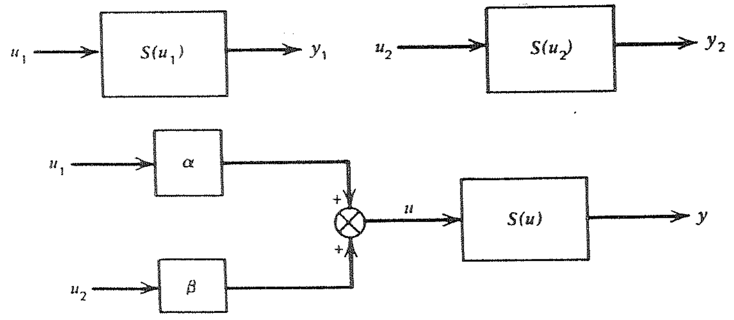


Figure 2 Diagrams for the principle of superposition.

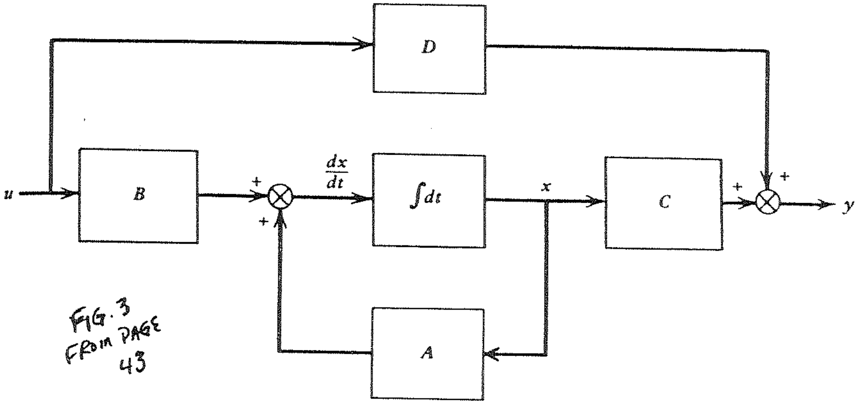


FIG. 3
FROM PAGE
43

Figure 3 Diagram for a generalized linear system.

A general model of systems with internal states can be obtained by expressing the time rate-of-change of the internal state as a function of the internal state and the applied input:

$$\frac{dx}{dt} = f(x, u)$$

$$y = g(x, u)$$

where x is the system state, u is the input to the system, and y is the output or the observable quantity of the system. For linear systems, these general system-state equations reduce to

$$\frac{dx}{dt} = Ax + Bu$$

$$y = Cx + Du$$

These linear equations look much more imposing when they are drawn in graphical form (Fig. 3). Equations of this form can be used to analyze many different systems with a spectrum of differing inputs.

Such mathematical equations have wide applicability, ranging from aerospace control systems and chemical process plants to models of economic and social systems.

The *raison d'etre* of systems design is optimization, and a class of models useful for this purpose are those referred to as "mathematical programming" or "mathematical optimization" techniques. All of these techniques incorporate some model of the system, along the lines we have just discussed, plus a mathematical statement of the criteria for optimality. This statement is called a value function or an objective function. The optimization process may attempt to maximize value, a benefit-cost ratio, output, reliability; or to minimize cost, risk, inputs; or some combination of these—whatever the system designer perceives to be the measure of the "best" system (Fig. 4).

The simplest, though inelegant, optimization technique is to examine all of the possible system alternatives, one-by-one, in an "exhaustive" search, and then to select that system or set of system parameters which produces the optimum value for the objective function.

Where defined, logical relationships exist between the system alternatives, all the analysis tools of mathematics may be brought into play—calculus, variational methods, Lagrange multipliers, and so forth. Elegant computer programs have been developed to search efficiently through large volumes of parameter space for optimum solutions.

Several new optimization techniques have been specifically developed to deal with the optimization problems of large systems. Linear programming models represent systems by a set of linear inequalities of the form:

$$\sum_j a_{ij}x_j \leqq b_j$$

and the objective function by an equation which is linear in the "decision" variables, x_j:

$$Z = \sum_j c_j x_j$$

The optimization process consists of selecting the x_j's to maximize or minimize Z subject to the constraints of the linear inequalities. The "simplex" method for the solution of linear programming problems was developed in 1947 by George B. Dantzig and his associates for the U.S. Air Force on Project SCOOP (Scientific Computation of Optimum Programs). Linear programming techniques were originally applied by the Air Force to such diverse areas as contract bidding; balanced aircraft, crew training, and wing deployment schedules; scheduling of maintenance overhaul cycles; personnel assignment; and airlift routing problems.

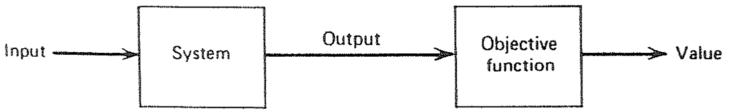


Figure 4 Mathematical optimization model.

Another technique, dynamic programming, was originally developed by Richard Bellman in the early 1950s at RAND. Dynamic programming models are a tool for analyzing multistage decision processes. One can imagine systems situations where a series of decisions must be made concerning a number of activities, or a time-sequenced set of decisions must be made concerning a single activity. Dynamic programming models make the following basic assumptions:

1. The returns from different activities can be measured in a common unit.
2. The return from any activity is independent of the allocations to the other activities.
3. The total return can be obtained as the sum of the individual returns.

These assumptions can be expressed as

$$\text{optimize:} \quad Z = f_1(x_1) + f_2(x_2) + \cdots + f_N(x_N)$$

subject to the constraint

$$x_1 + x_2 + \cdots + x_N = x_0$$

where the x_j's are the resources to be allocated and x_0 is the total resource available.

Often, systems problems can be modeled as a flow of something through the system. This flow may be automobiles through a traffic pattern, goods in a manufacturing plant, oil through pipelines, activities through a scheduling chart, or the like. These problems are analyzed with the aid of network flow models, where the network consists of interconnecting nodes and paths. One important technique is called PERT (Program Evaluation and Review Technique). PERT was developed by the U.S. Navy as a technique for preparing project schedules, and for assessing the progress of the project with respect to the schedule. PERT was first used on the Navy's Fleet Ballistic Missile Project—the Polaris project. The management of the Polaris project was complicated by the fact that there were many tiers of contractors and subcontractors. It was extremely difficult to understand the impact of problems in one area on other areas and on the overall project schedule. PERT made an important contribution to the resolution of these scheduling problems on the Polaris project.

There is a fundamental difference in the use of models in science and engineering. In science, a model is the final product. A model in science represents the distillation of all the knowledge about a certain phenomenon. In engineering, where the purpose is not codification of knowledge but the achievement of an objective, models are only a means to an end. How much modeling effort should a systems engineer undertake, and how detailed should the systems model be? This is a resource allocation problem. The systems engineer should continue to develop the systems model until the marginal benefit of improving the model falls below the marginal cost.

The Systems Viewpoint

Systems engineering is more than a knowledge and application of principles of systems design and systems modeling concepts. In what follows an attempt is made to give the reader a feeling for the point of view which makes systems engineering different from classical engineering.

The heart of the matter lies in the complexity of the system and in the danger of being unable to see the forest for the trees. The designer must somehow deal with the various subsystems and component parts in such a way as to optimize the cost/effectiveness of the overall system—which means avoiding the dangers of suboptimization. The word "suboptimize" was coined in 1952 by C. J. Hitch, and the following example is taken in part from his article.[2]

An excellent study, one of the classics of operations research, was performed during World War II on the optimum size of a merchant ship convoy.[3] The problem was the sinking of United States and Allied merchant ships by "packs" of German submarines in the North Atlantic Ocean. There is, of course, never enough data for such problems, because of the statistical variability in such things as sightings and sinkings and the numerous questions of luck and skill involved. However, the researchers were able to show (with what most system engineers would agree was reasonable confidence) that the number of merchant ships sunk when a convoy was attacked by a given pack of submarines was independent of the number of merchant ships in the convoy, but inversely proportional to the number of escort vessels

(such as destroyers) in the convoy. Furthermore, the number of submarines sunk in such an encounter was proportional to the number of escort vessels. It follows that the payoff, chosen as the ratio of submarines sunk to merchant ships sunk, varies as the square of the size of the convoy (assuming that the same ratio of escorts to merchant ships is retained). The recommendations from this study were put into effect, and the number of merchant-ship sinkings decreased drastically, contributing importantly to the winning of the Battle of the Atlantic, and consequently to the winning of the war. In fact, the decrease was even more dramatic than predicted; the submarines were so ineffective in the North Atlantic that they were transferred to more profitable missions elsewhere.

This celebrated problem has been the subject of a number of postwar studies, and it now appears that the change in tactics (increasing convoy sizes) was probably right, but for many of the wrong reasons. In fact, the study as described above is remarkable for the number of errors which have been made from the systems viewpoint.

In the first place, if one really believed the above conclusions, he would recommend taking every bottom available to the Allies and putting them into a single giant convoy. This is clearly ridiculous (in systems engineering, as in mathematics, extreme cases are often illustrative); the optimum convoy size must consider the disadvantages of increasing size as well as the advantages. The obvious disadvantages are that the convoy can move no faster than its slowest ship, and that the arrival of a convoy swamps port facilities, greatly increasing turnaround time.

In fact, the study is guilty of suboptimization; what has been optimized is the skirmish between a convoy and a submarine pack, for which the measure of effectiveness is the ratio of submarines sunk to merchant ships sunk. What should have been optimized is the Battle of the Atlantic, for which the proper measure of effectiveness is the goods delivered to the eastern shore of that ocean. Of course, one can quibble about modifications of this measure depending on the length of the war (do we want to maximize goods delivered during the next month or during the next year?) and the desirability of saving human lives (at least on our side), but the principle is clear: if the convoy is too large, it will take so long to assemble, load, sail, unload, and return that the amount of goods delivered may be considerably less even though we lose less shipping.

But even this viewpoint is a suboptimization, because the real objective is not so much to win the Battle of the Atlantic as to win the war.

---

[2] C. Hitch, "Sub-optimization in Operations Problems," *J. Op. Res. Soc. Am.*, vol. 1 (1953), 87–99.

[3] P. M. Morse, and G. E. Kimball, *Methods of Operations Research*, Wiley 1951.

And when the German submarines went elsewhere, they indicated that we had gone too far. It is a principle of competitive situations (and, to the extent that they can be considered games against nature, all stochastic systems) that when we have achieved our optimum strategy, we are indifferent as to what the enemy (or nature) may do. This concept is made formal in game theory. It follows that if the enemy (or nature) has a clear-cut preference available, we are not at the optimum. In this case, if the German submarines could clearly do better by leaving the North Atlantic, we must have made our convoys too large.

But even this is a suboptimization, because the objective of winning the war should be subordinated to the objective of optimizing the postwar world. In this example, such considerations would be stretching things; but it was, for example, a serious question in the decision to drop the bomb on Nagasaki. And of course, there are even higher objectives. So what is the system designer to do if he is several echelons farther down (e.g., designing an antisubmarine guided-missile system aboard one of the escort vessels)?

In answer, Hitch suggests "the relevancy of economics" which "involves the analysis of relations between suboptimizations at lower and higher levels." We would add that, while absorption in the problems of higher levels can lead to paralysis or, worse, severe political repercussions, the designer should always be cognizant of their problems and the effects of his actions on them. Most important, he should know the level of his own sponsor and select appropriate criteria with his sponsor.

As a rule of thumb, in addition to his own level, a systems designer should think one level up and one level down. He should think one level up because the task as he receives it is not completely defined (for the reasons discussed earlier) in that a problem cannot be completely formulated before it is solved. Similar arguments dictate that the designer needs also to think one level down.

The Compleat Systems Engineer

Finally, we turn to the man who must do all of this—the systems engineer. The use of the word "systems" is sufficiently pervasive that every large organization invariably has some people who are identified as systems engineers. Many of these people do not have, and need not have, an understanding of the broad range of systems concepts as they have been presented here. These people may have a well-defined, static role to fulfill in their organization; and the role, important as it may be, may have little requirement for these concepts.

What we mean by the "compleat" systems engineer is the man who is the creator, the innovator, the synthesizer of systems. This is the man who needs to have the "big picture"; who must see the path from systems requirements to systems operation; who can make decisions, implement ideas and bring the system into being.

Clearly, this man must be, in some sense, a generalist rather than a specialist. The ideal systems engineer is a "T-shaped man," broad, but deep in one field. His depth is provided by scholarly experience—a Ph.D. or equivalent—and the breadth by extended interests and abilities. Frequently he must become a "6-month expert" in a new field, such as meteorology or television or electroencephalography, but he will find that his background in mathematics and engineering will enable him to learn enough in a short time to allow him to work with real experts in the field.

In addition to systems engineering, he must know a good deal about administrative and marketing matters. In particular he must be a good salesman, because regardless of the merit of his ideas, he must convince some sponsor that his project is more worthy of support than the numerous other proposals which are invariably competing for the limited financing, equipment, or time available. He must know about the project system of management; he must know about costs and accounting procedures; and he must know about organizational and administrative politics, which probably cannot be learned from any book.

Of course, the man who knows all of this does not exist. In practice, the systems engineer does not need to know everything. What he needs to know is everything that is pertinent to his particular problem. In that sense, there are thousands of systems engineers who come remarkably close.

BIBLIOGRAPHY

Baumgartner, John S., *Project Management*, Irwin, 1963.

Bellman, Richard, *Dynamic Programming*, Princeton University Press, 1957.

Bryson, Arthur E., and Yu-Chi Ho, *Applied Optimal Control*, Blaisdell, 1969.

Chestnut, Harold, *Systems Engineering Methods*, Wiley, 1967.

Chestnut, Harold, *Systems Engineering Tools*, Wiley, 1965.

Dantzig, George B., *Linear Programming and Extensions*, Princeton University Press, 1963.

Ford, L. R., and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.

Goode, Harry H., and Robert E. Machol, *System Engineering*, McGraw-Hill 1957.

Hall, Arthur D., *A Methodology for Systems Engineering*, Van Nostrand, 1962.

Machol, Robert E. (Ed.), *System Engineering Handbook*, McGraw-Hill, 1965.

Machol, Robert E. (Ed.), *Information and Decision Processes*, McGraw-Hill, 1960.

Machol, Robert E., and Paul Gray (Ed.), *Recent Developments in Information and Decision Processes*, Macmillan, 1962.

Miller, Robert W., *Schedule, Cost, and Profit Control with PERT*, McGraw-Hill, 1963.

Schwarz, Ralph J., and Bernard Friedland, *Linear Systems*, McGraw-Hill 1965.

System Engineering Management Procedures. Air Force Systems Command Manual (AFSCM 375-5), United States Air Force, February 1964.

Wilde, D. J., and C. S. Beightler, *Foundations of Optimization*, Prentice-Hall, 1967.

4

Decision Analysis in Systems Engineering

RONALD A. HOWARD

Professor of Engineering-Economic Systems
School of Engineering
Professor of Management Science
Graduate School of Business
Stanford University

Ronald A. Howard is a Professor of Engineering-Economic Systems and a Professor of Management Science at Stanford University. He received his Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology and served on the faculty until 1965. He is the author of *Dynamic Programming and Markov Processes* and *Dynamic Probabilistic Systems*. He has editorial responsibilities for *Management Science*, *Operations Research*, *IEEE Transactions on Systems, Man, and Cybernetics*, *Journal of Optimization and Control*, and *Journal of Socio-Economic Systems*. He is the Series Editor for the Wiley series, Decision and Control. Howard's research and consulting interests are currently concentrated on decision analysis.

The past decade has seen the development of a new profession—decision analysis, a profession concerned with providing a rational basis for decision-making. While it may seem strange that people can make their living by helping other people make decisions, that is just what decision analysts do. So that we can better see the need for this new profession, let us start by taking a look at the kind of decision-making we use in our everyday lives (see Fig. 1).

Descriptive Decision-making

In this descriptive view of decision-making, we first examine the environment of human decisions. The environment can be described

Project Feedback, is an attempt in this direction. Its precursor was a study carried out by Stevens and Little for the Governor of Puerto Rico to find out how the very poor people of that island were feeling about the various attempts to improve their lot.[5]

Although we should recognize that this way of utilizing the methods of physical science has its severe limitations, we must also see that it has great potentialities, when carefully used.

BIBLIOGRAPHY

Ackoff, Russell L., and Maurice W. Sasieni, *Fundamentals of Operations Research,* Wiley, 1968.

Bellman, R., *Dynamic Programming,* Princeton University Press, 1957.

Cohen, Jacob W., *The Single Server Queue,* North Holland, 1969.

Cox., D. R., and H. D. Miller, *The Theory of Stochastic Processes,* Wiley, 1965.

Dantzig, George B., *Linear Programming and Extensions,* Princeton University Press, 1963.

Drake, A. W., R. L. Keeney, and P. M. Morse, (Eds.), *Analysis of Public Systems,* M.I.T. Press, 1972.

Jaiswal, N. K., *Priority Queues,* Academic Press, 1968.

Larson, R. C., *Urban Police Patrol Analysis,* M.I.T. Press, 1972.

Morse, P. M., and G. E. Kimball, *Methods of Operations Research,* Wiley, 1951.

Morse, Philip M., *Library Effectiveness: A Systems Approach,* The M.I.T. Press, 1968.

Morse, Philip M., *Queues, Inventories and Maintenance,* Wiley, 1958.

Morse, Philip M., and Laura W. Bacon, (Ed.), *Operations Research for Public Systems,* The M.I.T. Press, 1967.

Prabhu, Narahari U., *Queues and Inventories: A Study of Their Basic Stochastic Processes,* Wiley, 1965.

Takacs, Lajos, *Combinatorial Methods in the Theory of Stochastic Processes,* Wiley, 1967.

---

[5] J. D. C. Little, C. H. Stevens, and P. F. Tropp, "Puerto Rico's Citizen Feedback System," *Analysis of Public Systems,* A. W. Drake, R. L. Keeney, and P. M. Morse, (Eds.), M.I.T. Press, 1972.

7

Systems Engineering at the Jet Propulsion Laboratory

WILLIAM H. PICKERING

Director
Jet Propulsion Laboratory
California Institute of Technology

William H. Pickering is the Director of the Jet Propulsion Laboratory of the California Institute of Technology. He is a member of the National Academy of Sciences and a Charter Member of the National Academy of Engineering. Pickering has received the NASA Distinguished Service Medal, the Army Distinguished Civilian Service Medal, the Columbus Gold Medal, the Robert H. Goddard Memorial Trophy, the Crozier Gold Medal, the Spirit of St. Louis Medal, and the Italian Order of Merit.

The Evolution of JPL

I present here some thoughts based on practical experience with systems engineering at the Jet Propulsion Laboratory. Since 1940, JPL has developed from a graduate student thesis project in the aeronautics department of Caltech into the present organization with about 4000 members, 2000 of which are professional engineers and scientists working under contract to NASA to build and fly spacecraft to the planets. JPL has evolved from a purely research-oriented laboratory into one heavily engaged in the practical application of systems engineering of large and complex projects. In the process, we have developed for ourselves many of the principles leading to the successful application of systems engineering, and we have discovered how to organize an engineering team to accomplish a difficult project.

At the end of World War II, the Laboratory was working for the U.S. Army. It had been supported in the first years of the war by the Army Air Corps and given the task of understanding the principles of rocket motor design for application to aircraft problems. The Laboratory was successful in developing the jet-assisted takeoff (JATO) principle and was then asked to transfer its know-how to a commercial organization which would build JATO units in quantity (see Fig. 1). This phase of JPL's activity was essentially representative of engineering research. We were concerned with the engineering principles of the design of successful rocket motors, but very little with the problem of engineering these motors into an airplane, or of analyzing the application of these motors to an airborne mission.

Toward the end of the war, Army Ordnance asked the Laboratory to explore the application of rockets to long-range artillery applications. (In those days, "long range" meant 100 miles.) It was now necessary to understand a rocket system consisting of a rocket motor, fuel tanks, guidance, and payload. Accordingly, new types of engineers, such as aerodynamicists and electronic engineers, became part of the organization. The group began to function as a systems engineering team. Nevertheless, the emphasis was still on research and, because it was not yet necessary, very little systems engineering discipline emerged. While various engineering groups worked together on mutual problems and were responsible for many successful projects, by today's standards, they would be described as disorganized. The objective was not to optimize a design, but to build something that worked and to understand the engineering principles of the design. This, of course, is necessary before parametric studies leading to optimum solutions can be undertaken.

The next step in the evolution of JPL to a systems engineering organization was brought about by a request from Army Ordnance to develop an operational missile system. As there was a stated need to arrive at a production stage as soon as possible, it was agreed that, using the research rocket which had flown successfully, JPL would develop an operational missile system. This really meant that as many off-the-shelf items as possible would be put together to produce a workable device which, however, would be far from a well designed and engineered system. Indeed, this proved to be the case for this missile system, which was called the Corporal.

The system did work, and the military made it work even better, but it was expensive, inefficient, and required large amounts of support equipment. It pointed out the consequences of putting a system together rather than engineering the system.

From the Laboratory's point of view there were some valuable lessons to be learned:

Figure 1 The first jet-assisted takeoff (JATO) in the United States, August 6, 1941.

- If a complex physical system is to be operated by relatively unskilled personnel, the total system should be designed with a clear understanding of its end use and of the man-machine interface.
- The compromises inherent in bringing together devices designed for other purposes can only result in producing an inefficient system, difficult to operate and costly to buy and maintain.
- In order to assure an integrated and optimized system, design responsibility and authority for the complete system must be given to the implementing agency. There must be short communication channels between the various technical and engineering groups responsible for the development of the physical hardware and the project management which has the responsibility for attaining stated objectives. Then there can be reasonable assurance that, in optimizing the hardware to solve a technical problem, the overall

project objectives and constraints will be recognized and incorporated into the design.

- The transfer of knowledge from a developing agency to a producing industrial company is not simple. There are three major problems. First, it is practically impossible to document all of the important know-how involved in implementing complex hardware. It is too easy to fail to mention some procedure which is so much a part of your folklore that you just take it for granted. The second problem in transferring knowledge is that different organizations not only have different folklore but different methods of solving problems, of establishing acceptance criteria, of assuring quality. Third, the engineer who has developed the system finds that he must interpret his decisions to a group of people who have not been exposed to the effect that the system constraints have had on those decisions. Consequently the system documentation must be educational, as well as definitive, and the engineer will find himself involved in the educational process.

The Laboratory had an opportunity to show Army Ordnance that we could develop a better missile system than the Corporal, when we were asked to do the second generation Sergeant missile (see Fig. 2).

Figure 2 The Sergeant, solid propellant, tactical guided missile.

This assignment included the total system responsibility, with realistic user requirements and constraints. We developed the system, transferred it into industrial production, and assisted the Army in its initial field operations. Thus we were assigned a classical systems engineering task for a large and complex system. The project was successfully completed at just the time when we transferred to NASA in 1958.

Since joining NASA, the Laboratory has expanded its systems engineering capabilies. Our assignment for NASA is to conduct unmanned spacecraft missions to the moon and the planets. Hence we have developed and carried into practice a number of spacecraft systems which required advanced concepts and the welding together of many scientific and technical disciplines (see Fig. 3). Some of the systems considerations involved in these missions are discussed below.

Systems Concepts in Lunar and Planetary Projects

The capability for successfully accomplishing lunar and planetary missions was realized when five unique and relatively new space technologies were developed:

1. A launch vehicle for injecting a spacecraft into a trajectory designed to intercept the moon or a planet.
2. A spacecraft capable of operating unattended in space with a high order of reliability for periods of days or months.
3. An in-flight propulsion maneuver capability to provide the necessary target accuracy.
4. A system for determining the cislunar or interplanetary orbit of the spacecraft.
5. A system for communicating with the spacecraft: an uplink for commands and a downlink for data transmission and tracking.

These missions require systems consisting of perhaps a hundred thousand parts, several major systems contractors, many subcontractors, and projects requiring tens of thousands of man-years of effort. These projects often involve one- or two-of-a-kind designs, with cost and schedule constraints rarely permitting test flights. Thus the projects must be carried out in the presence of large unknowns, which include technical feasibility (always encountered in advanced designs) and the environments in which the systems must operate.

Systems Engineering

To carry out these projects with reasonable expectation of optimizing performance or of attaining project objectives within cost and schedule, a systems engineering approach is obviously necessary. Basically, the systems approach involves the optimization of the overall system as opposed to the piecemeal suboptimization of the elements of the system. This overall optimization is achieved in a number of steps:

1. Goal definition or problem statement.
2. Objective and criteria development.
3. Systems synthesis.
4. Systems analysis.
5. Systems selection.
6. Systems implementation.

The systems engineering of lunar and planetary missions—conceptualization, design, fabrication, operation—is carried out according to this scheme.

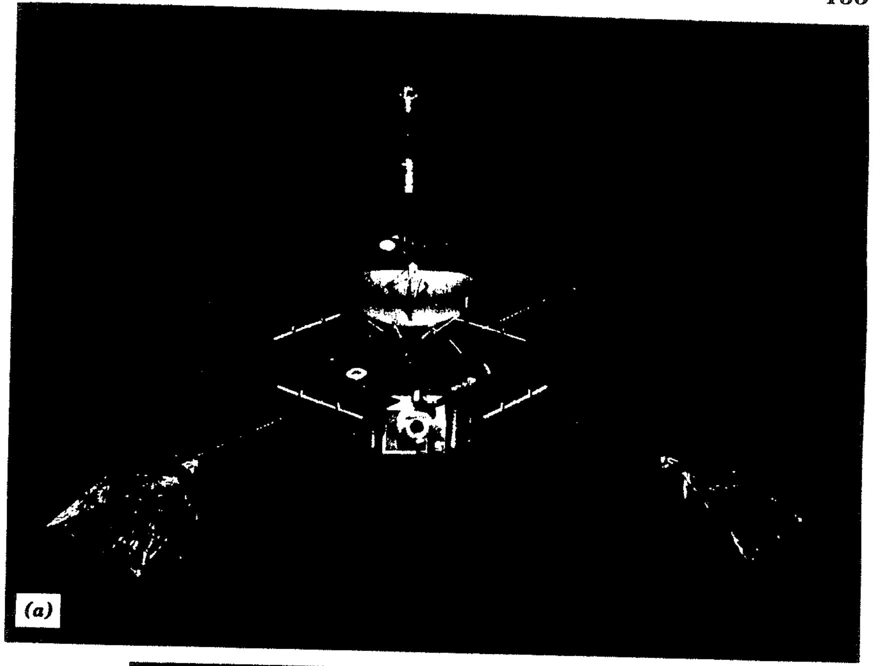
The goals of lunar and planetary programs are formulated in terms of a sequence of lunar and planetary missions, with objectives for each mission based on scientific and engineering feasibility studies. Design criteria provide a means of ranking alternative systems, establish the importance of various phases and objectives of the mission, and furnish a basis for making trade-off studies.

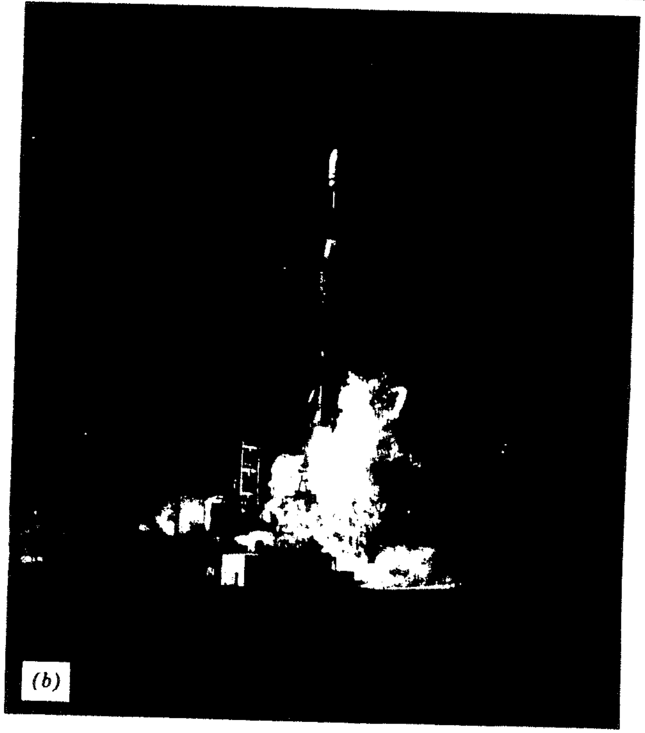
The systems synthesis and analysis is initially carried out in terms of functions required for the fulfillment of the mission. The design is then successively iterated for the effects of various hardware implementations. The selected design represents an optimum choice with respect to the mission objectives and the design criteria.

To focus and integrate the management and engineering efforts for a specific mission, a project organization is needed. Support from the total technical resources of the organization is provided by a "matrix" organizational structure. The classical organization chart with vertically aligned functional divisions is overlaid with horizontally aligned projects which intersect the division structure. An engineer from a functional division, working on a project, administratively reports to his functional division, but receives his work assignments from the project management. His performance is jointly reviewed by both the division and the project.

These projects must be broken down into comprehensible parts in order to be accomplished. The first major division of a space project is into systems. Lunar and planetary flight projects typically are composed of four systems (see Fig 4):

1. The Spacecraft System, consisting of the spacecraft and its support equipment.





2. The Launch Vehicle System, consisting of the launch vehicle and its support equipment.
3. The Tracking and Data System, which is responsible for the provision and maintenance of the earth-based tracking, telemetry, and command stations; the ground communications; and the operational facilities for the mission.
4. The Mission Operations System, comprising the management organization responsible for the design and execution of the mission operations.

Further breakdown of the systems into subsystems and components is made to reach a level of complexity that can be treated as a single element. Concurrent with the system breakdown, interfaces between the elements are established which define the functional boundaries of the elements.

This breakdown of a system into functional elements cannot be made arbitrarily. A great amount of managerial and engineering skill is required to select the interface topology, which affects the management control of a project, both administratively and contractually, and the engineering integration and operation of the system. From a management standpoint, interfaces are defined so as to optimize visibility and control, to isolate independently subcontracted elements, and to delegate authority and responsibility. From an engineering standpoint, interfaces are located so as to separate independent functions and to facilitate the integration, testing, and operation of the overall system.

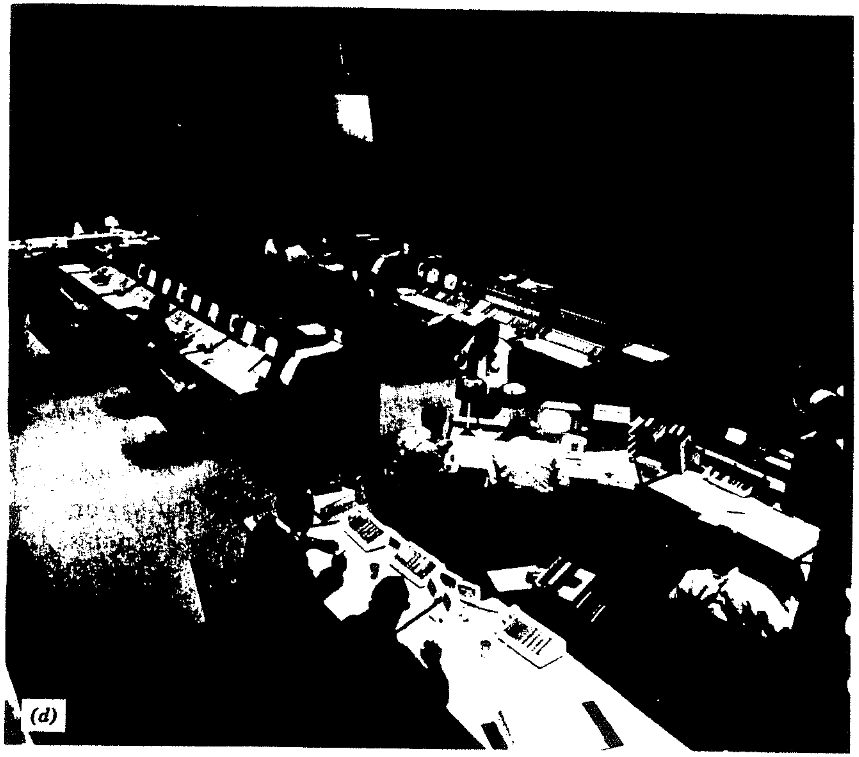


Figure 4 The four systems of a lunar or planetary flight project: *(a)* spacecraft system, *(b)* launch vehicle system, *(c)* tracking and data system, and *(d)* mission operations system.

In the early design stages, the exact mode of implementation for a system may not be known. For example, should the spacecraft be spin stabilized or inertially stabilized? Should power be obtained from solar cells or a nuclear device? Thus a system is initially defined in terms of performance specifications and constraints for each function, rather than in terms of the implementation schemes.

At this point, the design is defined by a set of subsystems, functions, and constraints for each subsystem, and an outline of the interface topology. Then the subsystem designs are projected to the point that the alternative subsystem implementations are understood. The subsystem performance characteristics and constraints are translated back through the system, and the process is iterated to produce an optimized, self-consistent preliminary design.

The preliminary design is specified as a set of functional requirements and interface control documents. These documents define the overall requirements and constraints levied on the spacecraft design by the mission, the major system interfaces, the subsystem interface topology, and the functions and constraints imposed by the system on the subsystem designs. They describe the design in sufficient depth to allow the detailed subsystem definition to proceed independently.

Design for Success

Reliability. Perhaps the most difficult requirement to satisfy in space missions is reliability. It presents the most challenging criteria for the management and engineering of deep-space systems.

There are typically tens of thousands of electronic parts in a planetary spacecraft design, many in themselves miniature assemblies (see Fig. 5). In the early 1960s, there was a serious concern that the sheer number of parts, each with its small but finite probability of failure, would result in an intrinsic unreliability at the system level that would render planetary flights unfeasible. Failures on early space flights tended to add credence to this concern. It took a fundamental change in the implementation of space projects before mission success occurred with regularity. It is now clear that the process of achieving reliability must start with the first concept of how a mission will be implemented and does not end until the last mission event has been successfully completed.

A Conservative Approach. It is necessary to take a very conservative approach to the design of lunar and planetary spacecraft in order to obtain the levels of ultrareliability required to achieve a high probability of mission success. The most conservative designs capable of fulfilling the mission requirements must be considered. This involves, wherever possible, the use of flight-proven hardware and, for new designs, the application of state-of-the-art technology, thereby minimizing the number of unknowns present in the design. New designs and new technologies are utilized, but only when already existing flight-proven designs cannot satisfy the mission requirements, and only when the new designs have been extensively tested on the ground.

SPACECRAFT PART COUNT COMPARISON

PART CLASS	MARINER IV	MARINER V	MARINER 1969	MARINER 1971
CAPACITORS	5,570	4,594	3,222	3,957
RESISTORS	15,607	10,781	9,916	11,031
DIODES	9,922	5,047	4,418	4,748
TRANSISTORS	4,323	3,027	3,035	3,296
IC's ACTUAL	–	594	2,763	3,063
(IC EQUIVALENT)	–	(17,226)	(80,127)	(88,827)
MISC*	3,798	1,225	1,105	1,256
ACTUAL TOTAL	39,220	25,268	24,459	27,351
(EQUIVALENT TOTAL)	39,220	(42,494)	(104,586)	(116,178)

*INCLUDES SWITCHES, CRYSTALS, FUSES, INDUCTORS AND TRANSFORMERS, RELAYS, ETC.

Figure 5 Spacecraft electronic parts count comparison.

Simplicity. Simple designs tend to minimize the number of parts, functional modes and interfaces, and have the fewest unknowns in terms of interactions with the system and with system operations. The philosophy of a fail-safe design should also be used; the spacecraft should be capable of performing a major portion of the mission even in the presence of failures. The system should have the capacity for operating in spite of some degradation in major parameters.

Redundancy. An important method for obtaining reliability at the system level is with redundancy. All mission-oriented functions should be backed-up by redundancies or alternate modes, and the system should be protected against failures in noncritical elements. Communications and power subsystems, being mission-critical elements, typically contain some element redundancy. On Mariner 1969, most of the computer events were backed-up by ground commands. In addition, there were more than a dozen cases of element redundancy.

Interface design. This is an extremely important part of the overall mission design. Interfaces exist between elements of the system, between systems, between people, and between organizations. The transfer of functions across interfaces tends to be particularly susceptible to design error. Hardware interfaces often also involve responsibility interfaces between different organizations. Thus it is essential that interface designs be simple in order that each side of the interface can be adequately designed and tested prior to mating, and in order that interface responsibilities can be clearly understood. On early Mariner spacecraft, the Spacecraft System/Launch Vehicle System interface occurred at the in-flight separation joint. On the Mariner 1969 spacecraft, the interface was moved down to the field joint. This eliminated the in-flight separation mechanism from the interface definition, thus simplifying the management interface as well as the testing and operations.

The design must also consider many other factors. To achieve the highest levels of confidence, the design must be capable of being tested and analyzed. Design confidence is obtained when test results correspond to analysis predictions.

Testing

The test program must be thoroughly integrated into the fabrication and assembly operations. The test results must be factored into the system design, and changes must be made where the test results indicate that mission objectives will be compromised.

The test program accomplishes many objectives. It provides data where additional information is needed to complete the design (development testing); it qualifies the design (type-approval testing); it validates the flight hardware for the mission (flight-acceptance testing); it produces calibration and signature data; and it provides training for the mission personnel (mission simulation). Testing, starting with the parts and materials, involves every level of assembly: components, subsystems, and systems.

The finite time available for testing restricts the range of operating states that can be investigated. The Mariner 1969 spacecraft with more than 10^{21} distinguishable states at the system level, even if sequenced through system states at the rate of one thousand states per second, could not be completely tested within the observed lifetime of the universe!

Mission Plan

Finally, the preparation for the mission is not complete until a detailed mission plan has been developed and tested, and the operations personnel have been trained in their duties. The mission plan must include an integrated, step-by-step account of the functions of all systems, through all phases of the mission. The mission plan should maximize the mission return in a reliable manner, consistent with the mission objectives and the constraints of the project systems.

Project Management

The transient nature of projects, the fact that no two projects employ the same resources, and the complex nature of the missions require that initially the mission objectives and the resources allocated to a project be specifically identified. Responsibility and authority must also be clearly delineated.

Lunar and planetary projects at JPL are managed in accordance with a management procedure specified in a Project Development Plan (PDP), which is prepared under guidelines provided in NASA management instructions. When management responsibility is assigned to JPL, the project management reports administratively to the Flight Projects Office at JPL, and technically both to it and to a counterpart program office within the NASA Office of Space Science.

Basically, the organization of JPL is focused on technical or professional disciplines. The majority of personnel and groups supporting project efforts within the Laboratory are members of various technical divisions. Each of these divisions assigns a full-time division representative to the project to assume responsibility for the efforts of the division for the project. Virtually all divisions, including service and support elements, participate to some degree in the activities of the project.

Project management must continually ask four questions: (1) will it work? (reliability), (2) will it operate as specified? (performance), (3) will it be ready? (schedule), and (4) for what cost? (resources). The responses to these questions concerning reliability, performance, schedule, and resources form the information base for a continual assessment of the status and progress of the project.

Project management interacts with the project elements through scheduled working meetings, which are conducted from the project initiation through to the conclusion of the mission. The project management interacts with NASA Headquarters, JPL management, and the project elements through a series of design reviews. Preliminary, design, and hardware acceptance reviews are conducted at system and subsystem levels. These reviews consist of presentations to a board, usually supported by back-up documentation in depth, followed by recommendations forwarded by the board to the project or concerned system manager. The general purpose of the reviews is threefold: (1) to bring independent and senior judgment, in the form of the review board, to bear on all aspects of the system or subsystem; (2) to assure consideration of the internal and interface characteristics by the appropriate managers and engineers; (3) to uncover and respond to residual or new problems.

The project attains its performance and reliability goals through the integrated stages of assembly, test, and assessment—starting with the procurement of parts and materials and culminating in the launch and encounter readiness reviews. The system performance is assessed by comparing test and analysis results against performance specifications. The reliability goals are achieved at the project level through an extensive reliability program involving many aspects of the project efforts.

To attain the reliability goals, there must be a parts qualification and control program, and the hardware fabrication process must be controlled. A quality assurance program must control hardware workmanship and assure that all test objectives have been met. There must be a configuration management program capable of identifying and verifying the specific components assigned to each spacecraft. An integrated test program is necessary, starting with the parts and materiel and providing functional and environmental tests of system elements at all levels of system assembly.

In addition, there must be a controlled problem/failure identification and resolution system capable of answering the following questions: (1) What failed? (2) How did it fail? (3) Why did it fail? (4) How was it fixed? (5) Why won't it happen again?

Finally, there must be a detailed mission plan, to be executed by qualified and trained personnel, with systems which are to be operated within tested and analyzed envelopes.

The timely management of project costs and schedule is extremely important, because unexpected problems invariably result in increased costs and schedule slippages in the absence of corrective action by project management. Project status must be continually reassessed: actual costs must be compared against planned costs, and progress against schedule.

Schedule control is absolutely essential to the successful completion of planetary missions. While it is possible to slip lunar flights on a month-by-month basis, opportunities for planetary launches occur at widely separated intervals—19 months between launch periods for Venus flights and 25 months for Mars flights. Thus, project problems must be identified and resolved without schedule slippage.

A control similar to the Mariner 1969 schedule milestone chart shown in Fig. 6 is used on most projects for summary reporting. It is not intended to be used for detailed planning because it may not

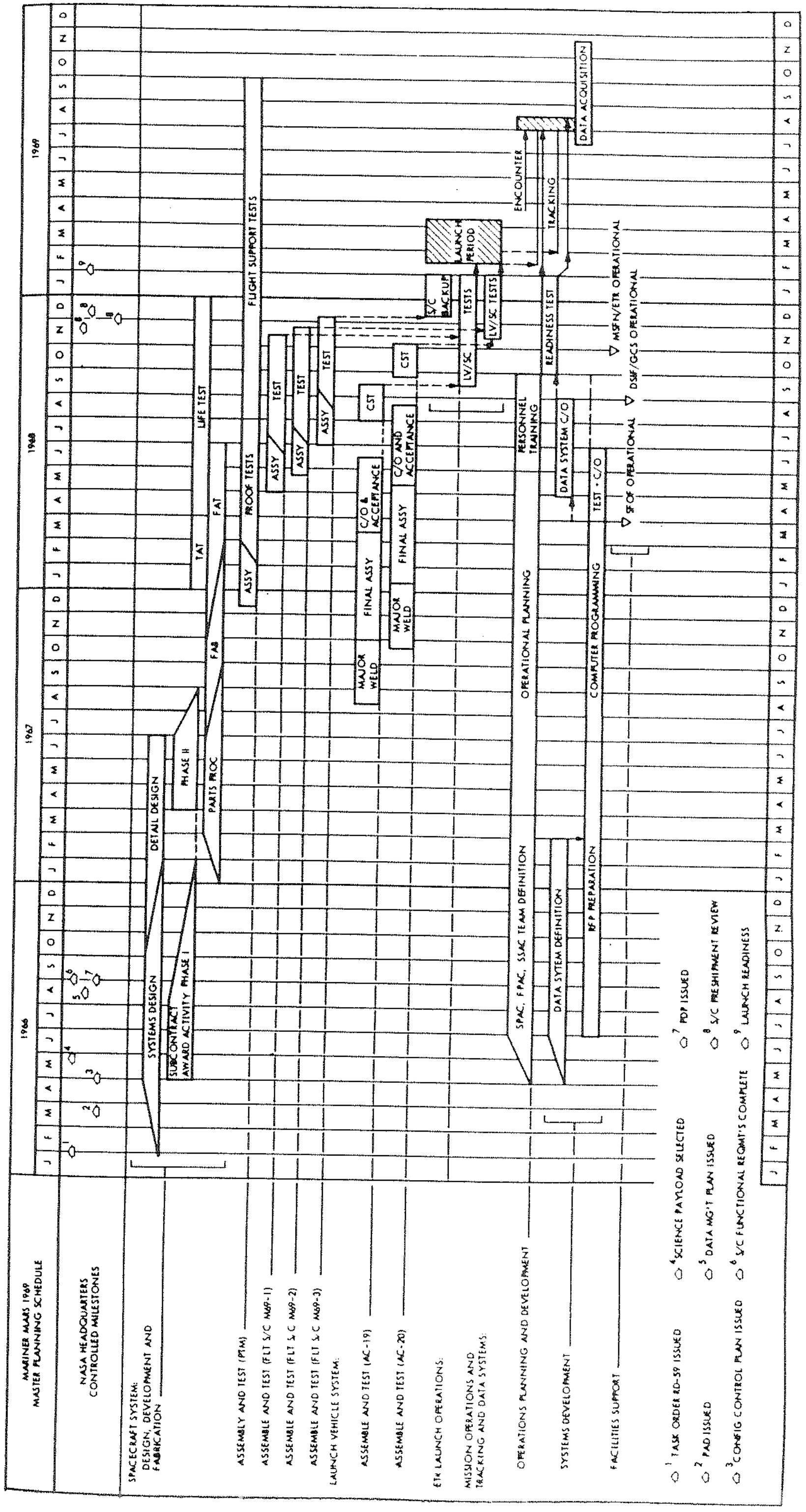


Figure 6 Mariner 1969 project master schedule.