Paper #139

Program (Systems) Engineering

Lynn E. Baroff, Robert W. Easter, Dr. Richard B. Pomphrey
NASA Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, MS 301-265, Pasadena, CA 91109
Lynn.E.Baroff@jpl.nasa.gov

Abstract

Program Systems Engineering applies the principles of Systems Engineering at the program level. Space programs are composed of interrelated elements which can include collections of projects, advanced technologies, information systems, etc. Some program elements are outside traditional engineering's physical systems, such as education and public outreach, public relations, resource flow, and interactions within the political environments.

Program Engineering supports decisions and directions of Program Management, from managing and analyzing program information to identifying priorities and imperatives for actions. We made a theoretical framework to model the major components of a Program and their inter-relationships. Applying the model to real-world programs yields benefits in visibility and integration of program elements throughout the Program lifecycle.

Like other engineering disciplines, Program Engineering applies scientific and mathematical principles to practical ends. This discipline, however, includes additional math (statistics, economics) and sciences (management, social sciences, psychology) to accomplish this practical work with the non-physical components of a Program. While this paper is really a process overview, with an explanation of how the discipline is being developed at JPL, we also promise future research results on the effectiveness of Program Engineering.

Program Engineering 101

INCOSE defines Systems Engineering in a practical and comprehensive way. It is described as "an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality… [Systems Engineering] integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs."

The meaning here is that Systems Engineering is focused, not just on the product being produced by the team of practitioners, but on the process of developing the product, and on the purpose of the product and the process by which it is used as well.

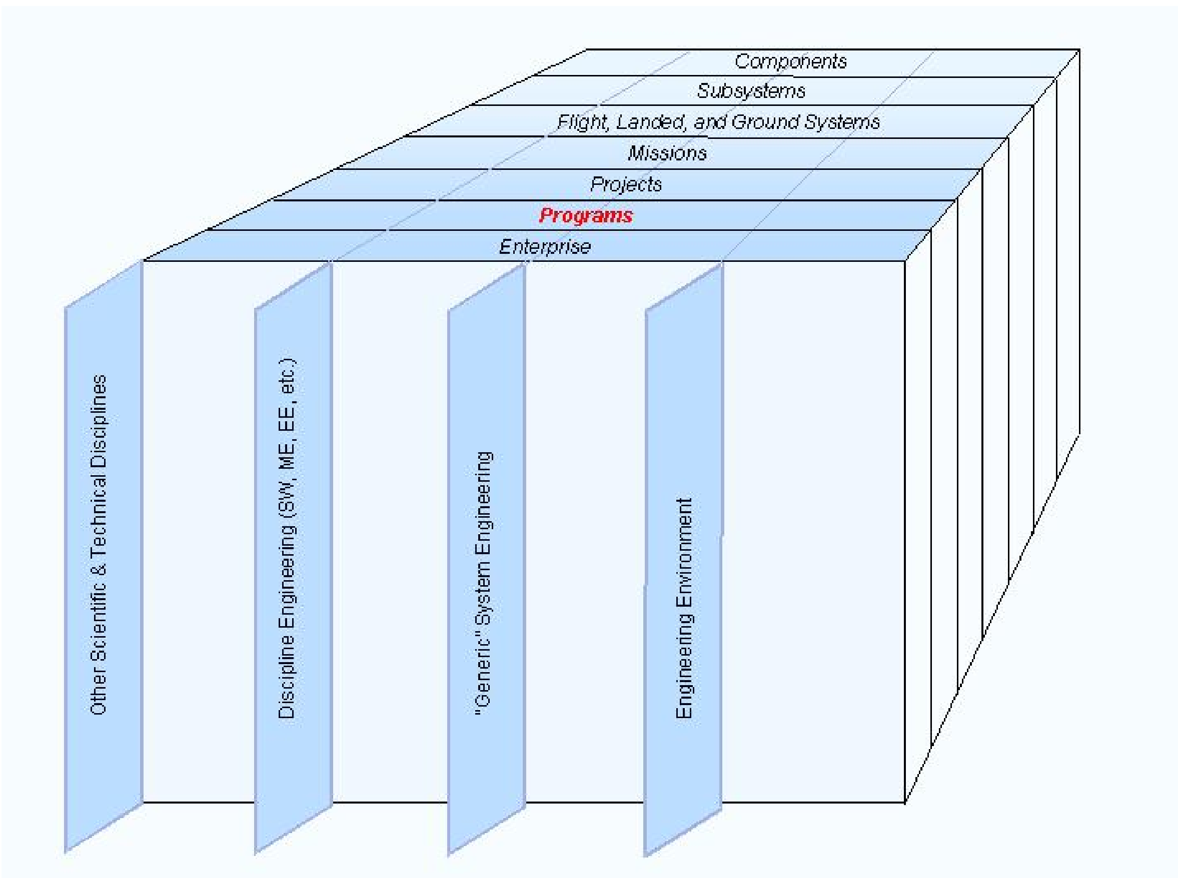
We understand *engineering* as application of scientific and mathematical principles to practical ends – such as the design, manufacture, and operation of efficient and economical structures, machines, processes, and systems.

A *system* is a group of interacting, interrelated, or interdependent elements forming a complex whole. The elements may be physical, such as a group of interacting mechanical or electrical components (like a sailboat or spacecraft); it might be a related set of operations, such as a communication or travel system (or the climate control system in your house); or it may be intangible, a set of ideas or processes, such as a social or political system (or the Scientific Method we all learn in school). It may even be a *project*, in which the system is a process that produces a product that is usable, and in which the system includes the product, the act of producing it, and the act of using it.

Systems Engineering, then, is applying the appropriate mathematical and scientific tools and principles to making a system and making it work – *wherever* it is in the grand scheme of things. Systems of various kinds are stacked within each other as the model here shows, all the way up to Enterprises (whole companies, government agencies, or themes of NASA scientific study). One level's system is another level's subsystem or component. Systems Engineering crosscuts the stacks, and so do other practices like the disciplines of engineering, the various sciences, and processes like collaboration and concurrent engineering.

Program Systems Engineering is not an entirely new discipline, but is partly a new application of the principles and methods of traditional systems engineering, at a level broader than a single product.

In our space exploration enterprise at JPL, we define a *Program* as "a long term continuing development or research activity that encompasses a defined area of interest or endeavor." Programs may be composed of (sub)programs, projects, missions, tasks, work units and other activities. At NASA and JPL, *Programs* are related series of undertakings which continue for a period of time (normally years), and which are designed to accomplish broad scientific, technical, or operational objectives. They usually include a family of related projects or missions, and many other components as well.



This description for Programs applies at a variety of agencies, but what each Program is intended to accomplish varies with the goals of each agency: Housing and Urban Development Programs have a different set of technical and operations objectives than Centers for Disease Control or Department of Defense Programs do. In the commercial world, the Program level (as those of us in NASA view it) might be comparable to a family of products produced by a commercial enterprise. The products may be tied together by associated purposes, like an integrated set of car care or personal hygiene products that *include the brushes and videos that demonstrate their uses.*

Our sailing example works here. Sailboats have components (sails, ropes, rudders and so on) grouped into functional subsystems and systems like propulsion, steering, galley, and so on; all contribute to the sailboat system as a whole. Construction of a sailboat is a project, and a trip you plan and execute from Marina Del Rey to San Francisco is a mission). Your sailing club is an enterprise – it runs programs like the Christmas parade or boats in the harbor, the annual series of races to various west coast ports, and the whale-watching field trip you conduct in Baja every spring for the UCSD Marine Biology students.

Programs as systems. Most of the typical principles of Systems Engineering apply to Program Engineering as well:

- *Boundaries:* Defining and understanding system boundaries, looking down from the next level up.
- *Interfaces:* Defining, understanding and managing the system's external interfaces.
- *Subsystems:* Partitioning the system into subsystems, defining and controlling the interfaces among subsystems, and applying the same principle in a consistent manner to elements lower in the hierarchy.
- *Requirements:* Utilizing consistent sets of system and subsystem requirements to help achieve objectives.
- *Metrics:* Defining consistent sets of performance measures and criteria to evaluate fulfillment of objectives.
- *Trade-Off Analyses:* Evaluating alternative formulation and implementation options against performance metrics and other value and/or utility functions.
- *Processes and Standards:* Defining processes and standards for repetitive activities, to create an environment—free-up time & resources—for human creativity and initiative.
- *Lexicons:* Creating and managing a standard set of terms and definitions, to assure consistent accurate communication.
- *Models:* Modeling the system and its subsystems – such modeling has a definitive role to play in understanding, optimizing and predicting the system's performance.

There are two important keys to understanding how a program works as a system. The first is to realize that only a few of the elements are physical products – the other elements are groups of people, other resources, and various processes. This idea isn't so unusual for system engineers. Our sailboat system, for example, needs more than wood and canvas to operate; it needs wind and people pulling ropes to make it go in the right direction and fulfill its purpose. Our spacecraft systems need stars to steer by (just like our sailors do).

This directs us to the second important key, also well known by system engineers. Whatever level system we are working on, it interfaces and interacts with other systems – and Program systems tend to be very open, with many interfaces that are *outside* the decided boundaries of the system itself. Our whale-watching trips with the UCSD students demand interactions with a lot of faculty and administrators, the parents, the Coast Guard, some specialty grocery stores, and so on. Our Planetary Exploration Program demands a lot of interactions with scientists around the world, their universities, news media across the country, the United States Congress, the Environmental Protection Agency, the Air Force, and on and on.

So Program Engineering demands more variety in the type of "mathematical and scientific principles applied to practical ends." Instead of stopping with physical sciences (physics, electronics, metallurgy) the way a spacecraft system engineer might do, program engineers may include the sciences that apply to these less tangible elements – social and behavioural science, economics, management, political science, and so on. The mathematics they will use will probably include statistics, risk and probability analysis, and very long term financial planning – besides calculus, geometry, and the complicated accounting of Earned Value Measurement.

Purposes of Program Engineering. The purposes of Program Engineering are to architect, direct, coordinate, and monitor the activities and elements of the Program, to ensure that the Program's intended objectives are realized by the Program system. Program Engineering is multi-disciplinary, and includes a few major functions:

- to integrate and validate the Program system;
- to solve problems related to system and element design and operation;
- to continually iterate Program system development and operation, to ensure that the system is able to meet its objectives;
- to support Program management in decisions related to Program system design and operation.

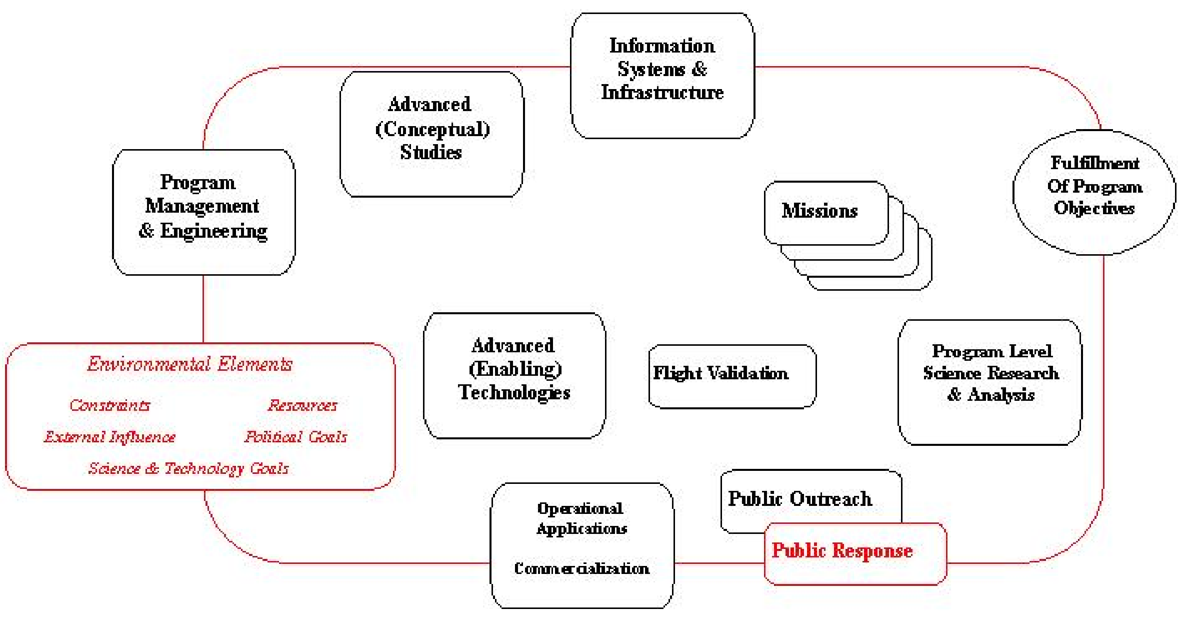
The primary outputs of Program Engineering are intellectual products and decision support, which are used by Program management to maintain balance and useful dynamics in fulfilling the objectives of the Program. (Program elements may have other, more tangible products; customers receive many of these, but some are mainly necessary to enable problem solving or decision-making by Program management.)

Program elements. Programs are more than collections of projects. Program element interfaces include public relations, links between funding and other resource flow over the long term, interactions within the local and national political environments, and others that are not part of traditional engineering's physical systems. From the highest perspective, all space exploration Programs contain much the same architectural elements:

- *Program Management:* those individuals who have cognizance and authority for all elements of a given program, including decision and funding authority for elements within the program.
- *Advanced Studies:* the identification and intellectual development of mission concepts that are candidates for implementation to help fulfill Program objectives. Advanced Studies are used to determine the feasibility of the mission concept from technical, cost and schedule perspectives, and are carried out to the point of final selection of a concept to be a mission.
- *Science Research and Analysis:* program-funded research and analysis of science data, which is directly applicable to fulfilling program objectives. In general, this refers to investigations that are beyond the scope of individual missions.
- *Information Systems and Infrastructure:* the information and logistical systems that support program work, including its missions. Infrastructure includes hardware, software and human resources required to provide a program's information system functionality.

- *Advanced (or Enabling) Technologies:* hardware, software and process technologies required to fulfill program objectives. "Enabling" technologies are those required to fulfil a set of mission requirements, or to deal with a program constraint. "Advanced" technologies are those not yet flight qualified (TRL 7 or below).
- *Missions:* the study, design, development and operation of a set of flight projects, which together are intended to fulfill the program objectives.
- *Operational Application:* the transfer and application of a functional capability to an operational system; especially a situation in which the program has provided a reliable proof-of-concept of the functional capability.
- *Public Outreach:* engagement of public interest, or the engagement of the interest of specific segments of the public, usually through communication of program information. This includes *commercial outreach*, which usually involves the transfer of technology developed for a program into the private sector.
- *Validation:* confirming that an advanced technology is qualified for flight, by providing objective evidence that the specific requirements for use are fulfilled. *Flight Validation* refers to performing the confirmation through actual flight.
- *Fulfilment of Program Objectives:* achieving program objectives, according to pre-determined success criteria.

These elements are illustrated in the un-architected model below, which shows Program elements external to the Program in red. Many of the elements shown, are *human systems,* and behave in accordance with different rules than physical systems do (Wheatley, 1992; Laufer, 1997)



Typical Elements of a Program

INCOSE has been paying attention to human aspects of mission systems – notice articles on "Organizational Architectures" and "Socio-Technical Systems" during 2003 in the "Systems Engineering" journal. Notice also an entire track of papers on "Human Factors in Mission Assurance" at the 2004 Space Systems Engineering and Risk Management Symposium sponsored by the Aerospace Corporation, NASA, and the U.S. Air Force Space and Missile Systems Center.

In developing a more formal practice of Systems Engineering at the Program (and Enterprise) levels, we hope to identify and *model* the relationships among these human elements – Program elements – in terms that can be consistent from Program to Program, and from one Program Engineer to another. For this reason, our Program Engineering discipline must apply (clearly, consistently) both qualitative and quantitative aspects of the social and behavioural sciences. In that way we will be able to explain (for one example) how Program Systems Engineering would tie together Advanced Technologies like nuclear electric propulsion with Public Outreach and Public Response, through the systematic use of communications media both internal and external to the Program.

Practicing Program Engineering

NASA space Programs, in accordance with NASA Procedures and Guidelines 7120.5B, are organized into two major phases, formulation and implementation. (Defence space acquisition Programs are organized the same way, but add a third phase, sustainment and disposal, to reflect the fact that most of these Programs result in an operational system that must be "used" over a long term. NASA Programs include operations as part of the implementation phase.) In our view, the activities of Program Engineering are different for Formulation than for Implementation.

Program formulation establishes success criteria for the Program; and defines the program concept (or architecture) and plan for implementation. During formulation, top-level requirements are identified and agreed upon, then incorporated into the Program Plan and passed on to the projects as objectives.

There are standard start-up activities common to all Programs that are part of the NASA and JPL space exploration enterprises. These activities are usually performed for the first time during the conceptual development of the Program – but Program formulation (or re-formulation) is usually ongoing throughout the life cycle of the Program.

Program Implementation activities in our space exploration world usually start with the formulation activities for the missions that are part of the program. Program Implementation includes designing and developing program products, measuring their effect (against the previously established success criteria), and iterating the program architecture.

Our view includes another division in Program activities, between "Architecting" and "Planning, Integration and Control." Our description of Architecture is the *purposeful arrangement* of Program elements – in other words, organizing the relationships among all those elements that make up a program, and defining the way the interactions among them work to satisfy Program objectives.

Planning, integration, and control are traditional management activities. Both Architecting and Planning, Integration and Control activities are practiced during both formulation and implementation phases of a Program.

The chart on the next page illustrates Program Engineering analysis, the abstract separation of the program into parts for individual study, and the study of the constituent parts and their interrelationships. Program functions, are categorized within two areas (Architectural, and Planning, Monitoring and Control) and within two program phases (Formulation and Implementation).

	Architectural Activities	*Planning, Integration and Control Activities*
Formulation Phase	✓ Identify or formulate Program goals ✓ Develop Program system architectural elements ✓ Analyze systems concepts and make feasibility trades ✓ Develop Program requirements ✓ Assess operational requirements ✓ Estimate cost and resources (conduct system trades in resources as required) ✓ Identify capability gaps ✓ Identify Program risks	✓ Partition Program into work elements, specify characteristics and interfaces among work elements ✓ Develop Program plans ✓ Develop success criteria and verify Program system architecture ✓ Design system evaluation, including reviews and review requirements ✓ LCC costing, O & M cost drivers ✓ Initiate risk management process ✓ Develop policies, procedures related to Program implementation ✓ Develop Program processes
Implementation Phase	✓ Define technical requirements (and Flowdown requirements) ✓ Assess & review system designs ✓ Maintain requirements trace ✓ Make Program architectural trades ✓ Rebalance risks ✓ Review problems where the Program is out of balance, or experiences difficulty in meeting requirements ✓ Manage the Program's change process	✓ Maintain Program Plan, performance measurement, reports, budgets, schedules, reviews… ✓ Monitor technical resources and resource margins ✓ Manage technical requirements ✓ Design, develop, sustain systems and technologies ✓ Deliver products; operate systems ✓ Document Lessons Learned ✓ Update and Assess Metrics ✓ Capture Knowledge

Program Engineering Tools

A variety of Program Engineering tools are in use or in development at JPL. Notice from these brief descriptions that they are very much like project concurrent engineering tools in their functions.

Availability analysis. In designing program concepts that have a requirement for continuous operations (e.g. a constellation of satellites or a Martian Robotic base) we have found it useful to apply "Availability" concepts. In particular the Aerospace Corporation's GAP (Generalized Availability Program) tool has been used to aid in determining robotic element reliability requirements and base resupply requirements to maximize the probability that a given operation can be carried out at any given time – and this is essentially the definition of "Availability". The GAP tool is also in use by Aerospace to analyze satellite programs.

Technology investment and Benefit/cost analysis: "Investment" perspectives have been found useful in developing program concepts and in making budget allocations for the type of programs engaged in by JPL, because of the relatively long term nature and relatively non-specific objectives of programs (when compared to projects) Charles Weisbin and others at JPL have developed a range of portfolio analysis tools to seek optimum allocations of budgets among technology development opportunities, based on future mission needs (Weisbin et al 2003). Simple benefit cost risk estimates (e.g. maximize benefit times probability of success divided by cost) have also proven useful in selecting program options in early conceptual stages

Program Engineering Notebook (PEN): PEN provides users (Program Engineers and Program Managers) with an integrated set of Program level data and analysis tools which can be used to architect, plan, integrate and monitor the Program System and its component elements. PEN can be used to conduct tradeoff and "what-if" analyses of various program architectures. Among the PEN functions are an integrated schedule and budget capability which allows a variety of movements and stretches of Program elements' schedule and budget.

Mission Reference List (MRL): The MRL provides the user with a comprehensive listing of all future space missions, together with overview information about each mission. The MRL is intended to include NASA and non-NASA, U.S. and foreign missions. It can be used as a high-level planning tool during either formulation or implementation, for either a Program or a Program element.

Flight Option Analysis Tool (FLOAT): FLOAT provides the capability of identifying and characterizing what flight opportunities may be available to a given flight experiment or payload.

Technology Information Management and Decision Support System (TIMDSS): TIMDSS provides the user with information on advanced technologies integrated with decision tools; this enables analysis of the use of selected advanced technologies for future spaceflight missions.

While the real measure of Program Engineering will be the efficacy of the various management decisions it supports, these tools (especially PEN) are expected to provide useful metrics on the processes of Program Engineering. The authors will be researching the effectiveness of Program Engineering as it is developed over the next few years, and will report back to interested INCOSE readers and practitioners regularly and with more thorough research results.

References

Jeffrey H. Smith, Julie Wertz, Charles Weisbin, "Building a Pathway to Mars: Technology Investment for Science Return." Journal of Space Mission Architecture, Fall 2003.

Margaret Wheatley, *Leadership and the New Science.* Berrett-Koehler Publishers, Inc., San Francisco, 1992.

Alexander Laufer, *Simultaneous Management.* American Management Association, New York, 1997.

Holly A. H. Handley and Alexander H. Levis, "Organizational Architectures and Mission Requirements." Systems Engineering, Volume 6, Number 3, 2003.

Andreas Gregoriades, Alistair Sutcliffe, and Jae-Eun-Shin, "Assessing the Reliability of Socio-Technical Systems." Systems Engineering, Volume 6, Number 3, 2003.

Lynn E. Baroff, "Human Factors in Mission Assurance – Cultural Influence on Mission Success and Risk." Proceedings of the 2004 Space Systems Engineering and Risk Management Symposium.

G. Rodriguez, Charles Weisbin, Robert Easter, "A New Method to Compare the Performance of Humans, Robots, and Integrated Human-Robot Systems in 21st Century Space Operations." JPL Publications, October 2001.

Charles P. Minning, David Crisp, "NASA's New Millennium Program: Flight Validation of Advanced Technologies for Space Applications." Journal of Space Mission Architecture, Fall 2003.

Biographies

Lynn E. Baroff is a Program Engineer in the Mission and Systems Architecture Section at NASA's Jet Propulsion Laboratory. When he first came to JPL 12 years ago, he was responsible for Management Development and Organizational Development, an internal version of a management consultant. He is one of the founding members of JPL's Project Formulation Support Team, and he represents the Team to both the Dawn and Terrestrial Planet Finder missions. Before coming to JPL in 1992, he was the General Manager of a management consulting firm, and engaged in what is now referred to as Enterprise System Engineering with clients including Xerox, Los Angeles County, Rockwell, and Toshiba. He is on the adjunct faculty of the University of Southern California, School of Public Administration.

Robert W. Easter leads a new JPL working group formed to apply Systems Engineering principles and practices at the Program and Enterprises levels at JPL, NASA and elsewhere. He is Staff Engineer of JPL's Systems Division, where he organizes program-level systems engineering for the Mars Exploration Program, New Millennium Program and other JPL Programs. He has been the JPL Representative to the NASA Exploration Team, an Agency-wide advanced planning activity responsible for developing long term Agency technology initiatives; and he has worked as the Director, JPL Systems Office, NASA Space Station Program Office.

Richard B. Pomphrey holds a Ph.D. in Astronomy. His career at NASA Jet Propulsion Laboratory spans a period of over twenty-five years in space exploration and research which covers a broad range of experience in: project and task management; systems engineering, design and analysis; information systems; advanced technology; mission development; space flight operations; and science and instrument support. This experience was acquired in support of Cassini, Galileo, Mars Pathfinder, Mars 94, Voyager, Viking and Infrared Astronomy Satellite (IRAS) Missions; Space Station, Earth Observing System (Eos), Astrophysics Data System (ADS),