

SYSTEMS ENGINEERING

*Design Principles
and Models*

Dahai Liu

 **CRC Press**
Taylor & Francis Group

WITH VITALSOURCE®
EBOOK



Accessing the E-book edition

Using the VitalSource® ebook

Access to the VitalBook™ ebook accompanying this book is via VitalSource® Bookshelf — an ebook reader which allows you to make and share notes and highlights on your ebooks and search across all of the ebooks that you hold on your VitalSource Bookshelf. You can access the ebook online or offline on your smartphone, tablet or PC/Mac and your notes and highlights will automatically stay in sync no matter where you make them.

1. **Create a VitalSource Bookshelf account at** <https://online.vitalsource.com/user/new> or log into your existing account if you already have one.
2. **Redeem the code provided in the panel below to get online access to the ebook.** Log in to Bookshelf and click the **Account** menu at the top right of the screen. Select **Redeem** and enter the redemption code shown on the scratch-off panel below in the **Code To Redeem** box. Press **Redeem**. Once the code has been redeemed your ebook will download and appear in your library.



DOWNLOAD AND READ OFFLINE

To use your ebook offline, download BookShelf to your PC, Mac, iOS device, Android device or Kindle Fire, and log in to your Bookshelf account to access your ebook:

On your PC/Mac

Go to <http://bookshelf.vitalsource.com/> and follow the instructions to download the free **VitalSource Bookshelf** app to your PC or Mac and log into your Bookshelf account.

On your iPhone/iPod Touch/iPad

Download the free **VitalSource Bookshelf** App available via the iTunes App Store and log into your Bookshelf account. You can find more information at <https://support.vitalsource.com/hc/en-us/categories/200134217-Bookshelf-for-iOS>

On your Android™ smartphone or tablet

Download the free **VitalSource Bookshelf** App available via Google Play and log into your Bookshelf account. You can find more information at <https://support.vitalsource.com/hc/en-us/categories/200139976-Bookshelf-for-Android-and-Kindle-Fire>

On your Kindle Fire

Download the free **VitalSource Bookshelf** App available from Amazon and log into your Bookshelf account. You can find more information at <https://support.vitalsource.com/hc/en-us/categories/200139976-Bookshelf-for-Android-and-Kindle-Fire>

N.B. The code in the scratch-off panel can only be used once. When you have created a Bookshelf account and redeemed the code you will be able to access the ebook online or offline on your smartphone, tablet or PC/Mac.

SUPPORT

If you have any questions about downloading Bookshelf, creating your account, or accessing and using your ebook edition, please visit <http://support.vitalsource.com/>

SYSTEMS ENGINEERING

*Design Principles
and Models*

SYSTEMS ENGINEERING

*Design Principles
and Models*

Dahai Liu



CRC Press

Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2016 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Version Date: 20150410

International Standard Book Number-13: 978-1-4822-8246-7 (eBook - PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

*To my wife, Juan, and my children, Meryl and Jenny,
for their love, patience, and support.*

Contents

Preface.....	xiii
Author.....	xv

Section I Systems and Systems Engineering Concepts

1. Introduction: Systems and Systems Engineering.....	3
1.1 Systems	3
1.2 Systems Classification	6
1.2.1 Natural Systems versus Man-Made Systems.....	6
1.2.2 Static Systems versus Dynamic Systems	7
1.2.3 Conceptual Systems versus Physical Systems	7
1.2.4 Open Systems versus Closed Systems.....	8
1.3 Systems Engineering	9
1.4 Brief History of Systems Engineering.....	12
1.4.1 From Reductionism to System Thinking	13
1.4.2 Early Practices	14
1.4.3 Government Role	14
1.4.4 Information Age.....	16
1.5 Summary.....	17
2. Systems Life Cycle and Design Processes.....	19
2.1 System Life Cycle	20
2.1.1 Operational Need	21
2.1.2 System Concept	23
2.1.3 Systems Concept Exploration and Validation.....	25
2.1.4 Engineering Model Development	26
2.1.5 System Production, Distribution, and Deployment.....	27
2.1.6 System Operations and Maintenance	27
2.1.7 System Phase-Out and Retirement	28
2.2 Systems Engineering Processes	29
2.2.1 Definition of Systems Engineering Process	29
2.2.2 Basic Concepts and Terminologies for Design Process....	30
2.2.3 Systems User Classes.....	33
2.3 Systems Engineering Design Processes.....	35
2.3.1 Conceptual Design	35
2.3.1.1 Identification of Needs	37
2.3.1.2 Feasibility Analysis	38
2.3.1.3 System Planning.....	41
2.3.1.4 Requirement Analysis	43

2.3.1.5	Functional Analysis at the Systems Level	45
2.3.1.6	System Specification Type A.....	49
2.3.1.7	Conceptual Design Review and Evaluation.....	50
2.3.2	Preliminary Design	51
2.3.2.1	Functional Analysis and Function Allocation.....	55
2.3.2.2	Design Tools and Analytical Models	58
2.3.2.3	Design Reviews	60
2.3.3	Detailed Design.....	61
2.3.3.1	Detailed Design Requirements and Specifications	61
2.3.3.2	CAD Tools and Prototypes in Detailed Design ...	62
2.3.3.3	Component Selection.....	64
2.3.3.4	Detailed Design Review.....	66
2.3.4	System Installation and Deployment, Operation, and Maintenance	67
2.4	System Engineering Design Process Models.....	68
2.4.1	Waterfall Model.....	69
2.4.2	Spiral Model.....	70
2.4.3	Vee Model.....	72
2.5	Summary.....	73

Section II Systems Methods, Models, and Analytical Techniques

3.	Systems Requirement Analysis.....	77
3.1	What Is a System Requirement?	78
3.1.1	Types of Requirements.....	80
3.2	Characteristics of Systems Requirements.....	81
3.3	Requirements Capture Technique.....	84
3.4	Requirements Analysis (RA).....	91
3.4.1	Affinity Diagram	92
3.4.2	Scenarios and Use Cases.....	94
3.4.3	Quality Function Deployment (QFD)	96
3.5	Requirements Management	100
3.5.1	What Is Requirements Management?	100
3.5.2	Why Requirements Management?	102
3.5.3	Requirements Management Using CORE.....	104
3.5.3.1	Requirements Management Software Tools ...	104
3.5.3.2	Requirements Management Example Using CORE.....	105
3.6	Summary.....	108
4.	Functional Analysis and System Design Models	111
4.1	What Is a Model?	112

4.1.1	Characteristics of Systems Models	114
4.2	Model Categorization	114
4.2.1	Classification Based on Model Format	114
4.2.1.1	Physical Models	114
4.2.1.2	Analogue Models	114
4.2.1.3	Schematic Models	115
4.2.1.4	Mathematical Models	115
4.2.2	Classification Based on the Nature of the Variables	116
4.2.2.1	Deterministic Models	116
4.2.2.2	Stochastic Models	117
4.2.3	Other Types of System Model Classification	117
4.3	System Design Models	119
4.3.1	Functional Models	121
4.3.1.1	What Is a Function?	122
4.3.1.2	Functional Flow Block Diagram (FFBD)	124
4.3.2	Functional Allocation	133
4.3.2.1	Resource Requirements for Functions	134
4.3.2.2	Allocation of TPMs	135
4.3.2.3	A CORE Example of Detailed Functional Analysis	138
4.3.3	Task Analysis Model	140
4.3.3.1	Input Requirements	141
4.3.3.2	Procedure	141
4.3.3.3	Output Product	141
4.3.4	Timeline Analysis Model	144
4.3.5	Link Analysis Based on the Network and Graph Model	144
4.3.5.1	Input	145
4.3.5.2	Procedure	145
4.3.6	Center of Gravity Model for Facility Location Planning	150
4.4	Summary	154
5.	System Technical Performance Measures	159
5.1	Technical Performance Measures (TPMs)	160
5.2	Systems Reliability	161
5.2.1	Reliability Definition	161
5.2.2	Mathematical Formulation of Reliability	163
5.2.2.1	Reliability Function	163
5.2.2.2	Failure Rate and Hazard Function	164
5.2.2.3	Reliability with Independent Failure Event	169
5.2.3	Reliability Analysis Tools: FMEA and Faulty Trees	178
5.2.3.1	Failure Mode Effect Analysis (FMEA)	179
5.2.3.2	Faulty Tree Analysis (FTA)	185
5.3	System Maintainability	192
5.3.1	Maintainability Definition	193

- 5.3.2 Measures of System Maintainability 194
 - 5.3.2.1 Mean Corrective Time (\bar{M}_{ct})..... 194
 - 5.3.2.2 Preventive Maintenance Time (M_{pt})..... 197
 - 5.3.2.3 Mean Active Maintenance Time (M)..... 197
 - 5.3.2.4 Mean Down Time (MDT)..... 198
- 5.3.3 System Availability 198
- 5.3.4 System Design for Maintainability 200
- 5.4 System Supportability 203
 - 5.4.1 Definition of Supportability 203
 - 5.4.2 Supply Chain and System Supportability 205
 - 5.4.3 Inventory Management: EOQ Model..... 207
 - 5.4.3.1 Ordering and Setup Cost 207
 - 5.4.3.2 Unit Purchasing Cost..... 207
 - 5.4.3.3 Unit Holding Cost 208
- 5.5 Human Factors and Usability Engineering 211
 - 5.5.1 Definition of Human Factors..... 212
 - 5.5.2 Work System Design..... 216
 - 5.5.3 Application of Anthropometric Data..... 218
 - 5.5.4 Usability Engineering 226
- 5.6 Summary 231
- 6. Decision-Making Models in Systems Engineering 237**
 - 6.1 Elements of Decision Making 238
 - 6.2 Decision-Making Models under Risks and Uncertainty 240
 - 6.2.1 Decision Making under Uncertainty 241
 - 6.2.1.1 Laplace Criterion 242
 - 6.2.1.2 Maximax Criterion..... 242
 - 6.2.1.3 Maximin Criterion 243
 - 6.2.1.4 Hurwicz Criterion 244
 - 6.2.1.5 Minimax Regret..... 244
 - 6.3 Decision Making under Risks..... 245
 - 6.4 Utility Theory 246
 - 6.5 Decision Tree 256
 - 6.5.1 Expected Value of Sample Information..... 261
 - 6.5.2 Expected Value of Perfect Information..... 261
 - 6.6 Decision Making for Multiple Criteria: AHP Model 262
 - 6.6.1 AHP Algorithms..... 265
 - 6.7 Summary 276
- 7. System Optimization Models 281**
 - 7.1 What Is Optimization? 282
 - 7.2 Unconstrained Optimization Models 283
 - 7.2.1 Derivatives and Rate of Change 283
 - 7.3 Constrained Optimization Model Using Mathematical Programming 291

7.3.1	Linear Programming.....	292
7.3.1.1	Formulation.....	293
7.3.1.2	Solving LP Models Using Graphical Method....	296
7.3.1.3	Simplex Algorithm.....	302
7.3.2	Solving LP Using a Spreadsheet.....	310
7.3.2.1	Step 1. Activation of Solver Function in Excel ...	310
7.3.2.2	Step 2. Set-Up of LP Model in Excel	312
7.3.2.3	Step 3. Solving the LP Model.....	312
7.4	Summary	313
8.	Process Modeling Using Queuing Theory and Simulation	319
8.1	Basic Queuing Theory	320
8.1.1	Queuing System	320
8.1.1.1	Arrival Process	321
8.1.1.2	Queue.....	322
8.1.1.3	Service Process	323
8.1.2	M/M/1 Queuing System.....	325
8.1.2.1	Exponential Time Arrival and Poisson Process	325
8.1.2.2	Birth-Death Process	328
8.1.2.3	Measures of M/M/1 Queue.....	333
8.1.3	Multiserver Queuing Systems	337
8.1.4	Queuing Systems with Finite Population.....	341
8.2	Application of Queuing Theory.....	345
8.3	Queuing System Analysis Using Discrete Event Simulation (DES) ...	347
8.3.1	Introduction to Simulation	348
8.3.2	Discrete Event Simulation (DES)	349
8.3.3	Data Analysis Using Goodness of Fit	350
8.3.4	DES Model Using Arena.....	353
8.3.4.1	Entity.....	354
8.3.4.2	Attribute	354
8.3.4.3	Resources.....	354
8.3.4.4	Queue.....	354
8.3.4.5	Variable	354
8.3.4.6	Step 1: Generate Parts Arrival and Assign Parts Attributes.....	357
8.3.4.7	Step 2: Drilling Process	358
8.3.4.8	Step 3: Decide for Rework	359
8.3.4.9	Step 4: Collect Time in System and System Exit.....	360
8.3.4.10	Step 5: Execute the Model and Obtain Results ...	361
8.4	Summary	361
9.	Engineering Economy in Systems Engineering.....	367
9.1	Interests and Time Value of Money.....	368

9.2	Economic Equivalence.....	369
9.2.1	Present Value (P) and Future Value (F).....	370
9.2.2	Annual Value (A) and Future Value (F)	371
9.2.3	Annual Value (A) and Present Value (P).....	373
9.3	Decision Making Using Interest Formula	376
9.3.1	Present Value, Future Value, and Annual Value Comparison	378
9.3.1.1	Present Value Equivalence (PE) Criterion.....	378
9.3.1.2	Future Value Equivalence (FE) Criterion	380
9.3.1.3	Annual Value Equivalence (AE) Criterion.....	380
9.3.2	Rate of Return.....	381
9.4	Break-Even Analysis	384
9.4.1	Cost Volume Break-Even Analysis	384
9.4.2	Rent-or-Buy Break-Even Analysis	385
9.5	Summary	386

Section III Systems Management and Control Methods

10.	Systems Management and Control.....	391
10.1	Systems Management Planning.....	391
10.1.1	Engineering Design Project Management and Design Teams	391
10.2	Systems Engineering Management Plan (SEMP).....	393
10.2.1	Cover Page.....	393
10.2.2	Table of Contents	394
10.2.3	Scope of Project.....	394
10.2.4	Applicable Documents	394
10.2.5	Systems Engineering Process	394
10.2.6	Transitioning Critical Technologies.....	397
10.2.7	Systems Integration Efforts.....	397
10.2.8	Additional Activities.....	399
10.2.9	Appendices.....	399
10.2.10	Systems Engineering Master Scheduling.....	399
10.3	Systems Control Models.....	400
10.3.1	Critical Path Method (CPM).....	400
10.3.2	Program Evaluation and Review Technique (PERT).....	407
10.4	Summary	411
	Bibliography.....	413
	Appendix I.....	419
	Appendix II	435
	Appendix III	437
	Index	449

Preface

With more complex systems being designed and utilized, understanding concepts of systems and mastering systems engineering methods are of utmost importance for systems designers. Complexity exists in all levels of systems and their subsystems, even components, which makes the design challenging. Guided by systems science, systems engineering applies scientific theories and models and incorporates the factors of system elements, including humans, machines, and environments, into unified models to analyze the effects of various system elements on system behaviors, which, in turn, seek the optimal course of actions to design these complex systems. Systems engineering is multidisciplinary in nature, integrating concepts, models, and techniques from a number of disciplines, including natural science, social science, and engineering, at both basic research and applied research levels. For the past several decades, systems engineering has grown rapidly in its scope and application and shown significant benefits for the design of large, complex systems.

I have been educating systems engineers for over ten years, teaching systems engineering at both undergraduate and graduate levels. My students are different from most systems engineering majors, as these undergraduate and graduate students have backgrounds in a large variety of areas, including psychology, physiology, and some engineering fields. With my experience teaching students with a variety of backgrounds, I feel that a textbook in systems engineering targeted at nontraditional engineers is in demand. The current systems engineering textbooks are either too technical or at a high conceptual level, giving these students very limited choices. Students need a good book that not only gives them exposure to the concepts of systems and systems engineering, but also enough technical expertise for them to use and apply what they learn on the job. That is the rationale for developing this book.

This book is written primarily for students with diverse backgrounds to learn about systems and systems engineering, and, more specifically, to be able to use and apply the models and methods in the systems engineering field. The materials included in the book have been taught for many years in the Human Factors and Systems Department at Embry-Riddle Aeronautical University; it has integrated feedback from students and colleagues and is written at a level appropriate for those groups of students to learn systems engineering, especially those nonengineering students who have no prior exposure to this subject. Engineering students, on the other hand, may also find this book useful and handy, as it provides a comprehensive overview of the subjects as well as the relevant analysis models and techniques. This book should serve well as a reference book for professional systems engineers.

This book has ten chapters, organized in three parts, including systems and systems engineering concepts (Chapters 1 and 2), systems methods, models, and analytical techniques (Chapters 3 through 9), and systems management and control methods (Chapter 10). The approach leans toward process-oriented and model-based systems engineering, with necessary topics covered in different chapters. Chapter 2 describes the system life cycle and systems engineering design process. Systems design starts with requirements; beginning with a good set of requirements is critical for the success of the design, and developing and analyzing requirements are not usually covered by a systems engineering book; that is why Chapter 3 is dedicated to this subject. It is believed that students will have a solid understanding of system requirements after having read this detailed information; a software package (CORE) is also used to introduce the requirement management process. Chapter 4 describes the design process in greater detail, especially the functional models. Chapters 5 through 9 explain the necessary technical measures and models that most systems designs will involve, including reliability, maintainability, supportability, usability (human factors), decision-making models under risks and uncertainty, optimization models, process models (queuing models and simulation with Arena software), and cost analysis using engineering economy models. The last chapter (Chapter 10) gives a comprehensive overview of systems management based on systems engineering management plans (SEMP) and systems control based on critical path method (CPM) and program evaluation review technique (PERT) models. In terms of the mathematics involved, it is believed that no specific prerequisite requirements are needed, except for fundamental algebra and basic probability and statistics theory; Appendix I is provided at the end of the book for readers to review these materials. This book can be used as a textbook at either undergraduate upper-level or graduate starting-level courses; exercise problems are provided at the end of each chapter for students to gain hands-on experience with the concepts.

I thank each individual who has encouraged me and assisted me in the process of writing this book. Specifically, I would like to thank Meaghan Hart, Julian Archer, and Hemali Virani for their assistance in preparing this book. Special thanks go to the senior editor Cindy Carelli, project coordinator Laurie Oknowsky, project editor Richard Tressider, and Michele Smith of CRC Press for their assistance, advice, and support throughout the process. I would also like to thank Katie Thacker and Bethany Maddox of Vitech Corporation and Jon D. Santavy of Rockwell Automation for their cooperation with the project. And finally, I want to express my appreciation to my family; their love and support are the inspiration for my career.

Dahai Liu

Author



Dahai Liu is currently a professor in the Department of Graduate Studies in the College of Aviation at Embry-Riddle Aeronautical University, Daytona Beach, Florida. He has been teaching systems engineering since 2002. His research areas are system modeling and simulation, human factors engineering in unmanned systems, human/machine learning, and artificial intelligence. Professor Liu has over 70 publications in professional journals and proceedings of conferences, technical reports, and a number of book chapters. He has conducted numerous research and consulting projects funded by the government and industrial companies such as the National Aeronautics and Space Administration, the Office of Naval Research, the Federal Aviation Administration, and Honeywell. Professor Liu received his Ph.D. in industrial and management systems engineering from the University of Nebraska at Lincoln in 2002 and his M.S. in systems engineering from Tsinghua University in 1996.

Section I

Systems and Systems Engineering Concepts

1

Introduction: Systems and Systems Engineering

Systems engineering and systems science has become one of the most important, comprehensive, and fundamental fields, and has wide application in almost every area of our society; from the macromanagement of government to the daily production of manufacturing facilities, from the development of spacecraft to the design of small consumer products, systems engineering is being applied at all times, at different levels. As one of the engineering disciplines, systems engineering studies systems; in this first chapter of the book, we will review the basic scientific concepts of systems, understand the background for systems science, and examine the evolution of systems and systems engineering from a historical perspective. This will help readers to understand the origin of systems science and the need for systems engineering. In this chapter, we will first define systems and systems engineering, describe the unique characteristics of systems engineering, and give a brief historic introduction to systems engineering development.

1.1 Systems

The term *system* is no stranger to us; we have been exposed to different kinds of systems since our early school years. According to the *Oxford English Dictionary*, a system is “a set of connected things or parts forming a complex whole,” or “a set of principles or procedures according to which something is done; an organized scheme or method.” Although the basic definition does not change, the term means slightly different things in different disciplines; for example, in physiology, human systems are sets of organs working together to serve common human physiological functions, such as the human nervous system, which serves as the body control center and communications network for the human body; or the human digestive system, which turns food into the energy humans need to survive and process food residue and waste for disposal. For human factor professionals, a system is the combination of human, machine, and

the surrounding environment. For systems engineering, we must define “system” more specifically:

A system can be broadly defined as a set of integrated components that interact with each other and depend upon each other, to achieve a complex function together. A system can be decomposed into smaller subsystems or components and a system may be one of the components for a larger system.

The following characteristics are present in all systems:

- *A system has a main function or a meaningful purpose.* To perform a function, a system takes input and generates output through the mechanism defined in the system structure. Systems’ functions, functional models, and analyses are illustrated in greater detail in Chapters 3 and 4. In this chapter, we briefly introduce the function’s concepts. For example, an automobile system’s function is to transport humans/goods to destinations. To perform this function, the system receives inputs and energy that allow the driver to start, control and drive the system get to the destination location; its inputs include the driver’s control and energy provided by the gasoline or electricity, and the output of the system is the velocity change (momentum).
- *A system has a hierarchical structure.* As seen in the definition, a system usually consists of many subsystems that can also be smaller-scale systems; meanwhile, a system is usually a subsystem of a higher-level system. For example, our planet is a large biological system, consisting of many smaller systems, such as the ocean and continents. Planet Earth is simultaneously a subsystem of the larger solar system.
- *A system has interacting subsystems and components that interact with each other.* Components’ interactions are necessary for achieving the system’s functions. A system is not simply the sum of all components, but rather it is an integrated whole package of the subsystems and components working in a well-defined hierarchical structure.
- *A system has a life cycle.* The system life cycle is the time span that starts from the concept of the system, continuing through the exploration of the concept, design and development, system operation, and maintenance, until the system is retired and discontinued.
- *A system has reliability.* No system is 100% reliable and error free. As Murphy’s law states, “Anything that can go wrong will go wrong” at some point. One of the most important design objectives is to keep Murphy’s law in mind and make the system more reliable.

More formally presented, systems have the following elements:

1. Subsystems and components: These are the fundamental constructs/functional units for systems. If desired, systems can be decomposed almost infinitely, all the way to the microworld level, such as electrons and atoms. For most design purposes, it is not necessary to decompose a system to this level of depth. As a rule of thumb, the decomposition usually stops at the assembly level, that is, at the level at which the commercial off-the-shelf (COTS) items can be obtained externally. The lowest level of assembly components and the other levels of the system are called subsystems. Subsystems are described through a hierarchical numbering system such as 2.0, 2.1, 2.1.1; by definition, 2.1 is a subsystem of 2.0, and 2.1.1 is a subsystem of 2.1. The hierarchical modeling approach will be illustrated in greater detail in Chapter 4. It is important to note that the subsystem and components may sometimes not be tangible hardware items. Depending on the nature of the system, some components can also be software, human, or even information.
2. Systems components have attributes. These attributes, often called design-dependent parameters (DDP), define and specify the systems components. For example, the physical dimensions of a component, mean time between failures (MTBF), power input and output, and so on. The purpose of systems design, to some extent, is to derive these quantitative and qualitative attributes from the systems requirements, so that specific system components can be built or obtained.
3. Systems components and subsystems interact and regulate system behaviors through different relationships. A system starts with user/customer requirements; *requirements* are the *basis* for systems *functions*; higher-level *requirements* can be *refined* by lower-level *requirements*; higher-level *functions* are *decomposed* by lower-level *functions*. Each *function* is *performed* by one or more *components*. These relationships are essential for successfully translating the systems requirements into component attributes, providing system rationale, and providing traceability for systems design activities. Systems design and modeling, including the specification of the system components and their relationships, is discussed in greater detail in Chapters 3 and 4.

Let us illustrate the systems concepts above with the example of a fixed-wing airplane (such as a Cessna 172). The function of a fixed-wing airplane is flight. In order to achieve this function, an airplane needs to have the necessary components; these components are used to construct the airplane, and they work together and interact, for the airplane to be able to take off, cruise, and land safely. Meanwhile, the components must be able to communicate with other airplanes and air traffic controllers. Among fixed-wing airplanes, major components are the airframe and propulsion systems. The airframe

subsystem can be further decomposed into smaller subsystems, such as fuselage, empennage, wing, landing gear and so on. The empennage subsystem consists of stabilizers, rudders, tabs, and elevators. Propulsion systems consist of components such as the power-plant subsystem (the engine) and electrical control systems (avionics). These smaller components are themselves a system on a smaller scale, which can also be further decomposed into smaller components, if desired. All the components work and interact with each other to provide some kind of function, in order to serve the bigger system's functions.

1.2 Systems Classification

Depending on the perspectives from which a system is studied, it can be classified into different categories. Understanding the system categories can help us to narrow the scope of the systems and derive common system characteristics. In categorizing the system, one has to keep in mind that none of the classifications makes a clear cut, and furthermore, any subsystem might belong to several different categories; for example, a man-made system could be dynamic and closed-loop controlled. Generally speaking, a system can be classified into one or more of the following categories: natural or man-made system, static or dynamic system, conceptual or physical system, and open or closed system.

1.2.1 Natural Systems versus Man-Made Systems

A natural system is a self-organized system that nature formed after millions of millions of years' selection and development. Examples of natural systems are the planet, oceans, and natural lakes. A natural system sustains itself by self-organizing to a state of equilibrium, for example, the food chain in a natural lake. Any disturbance to this equilibrium can be devastating for the natural system. A man-made system, on the other hand, is made by humans. Man-made systems, such as computers or automobiles, cannot be obtained from nature, but only through the creative efforts of humans. Systems engineering studies man-made systems as objects, with less concern for natural systems. One has to keep in mind, however, that there is no absolute isolation between natural systems and man-made systems. As a matter of fact, man-made and natural systems constantly interact with each other and sometimes make a huge impact on each other. Man-made systems often need inputs from nature (i.e., the automobile needs gasoline, which is made from crude oil from the natural world) and rely on nature to process the waste generated (e.g., the greenhouse gases from the automobile). With more advances in our technological development, man-made systems are becoming more complex and powerful, which requires more resources from Mother Nature. Meanwhile, more waste returns back to

nature, often polluting and causing the natural environment to deteriorate. Industrial pollution is becoming a more serious problem now, and with more awareness from humans, environmentally friendly systems designs have become part of system competitiveness and advantages.

1.2.2 Static Systems versus Dynamic Systems

Systems can also be classified as *static* or *dynamic*. Static systems are those structural systems that do not change their state within a specified system life cycle, such as a bridge, a building, or a highway. A dynamic system is one where its state, or the state of its components, changes over time, either in a continuous or discrete manner. A dynamic system's state can be considered a function of time, its change taking place at either a more deterministic rate or a more stochastic rate (think of a large service center's change of state as customers arrive in a random pattern). Similarly to natural versus man-made systems, the distinction between static and dynamic systems is relative, not absolute. Depending on the different perspectives from which a system is being analyzed, a system may be considered static or dynamic. For example, from Earth, within a short period of time, the Great Wall of China is considered a static system; however, if one observes it from outer space, the Great Wall is certainly moving, although at a very slow speed. While there may be no absolutely static systems, within the context of systems engineering, we treat some fixtures or structures as static. By doing so, we can simplify the problem, as we need to focus only on their dynamic components. For example, when we study operations within a production facility, the facility building is considered static; we only need to concentrate on the dynamic aspects of the system, such as material flows and human activities to investigate the effectiveness of production management.

1.2.3 Conceptual Systems versus Physical Systems

Our world comprises physical systems, and physical systems consist of objects that can be seen, touched, and felt. Natural systems such as animals, bacteria, lakes, and humans are all physical systems; physical systems also include man-made systems, such as the computers, appliances, tools, and equipment that we humans use on a daily basis. Conceptual systems are those consisting only of concepts, not real objects, so we cannot visually see or physically touch these systems. Conceptual systems illustrate the relationships among objects and allow us to understand the system and communicate details about the system's structures and mechanism. A simulation model of a factory operation process, a blueprint of the machine assembly, or the information processing model of human cognition and perception would be examples of conceptual systems. Science and mathematics are the fundamentals of conceptual models. A conceptual model can be general for a wide range of objects or it can be specific for a certain physical system. In systems engineering, conceptual

modeling serves as a basis for the physical system. Before the actual physical systems are manufactured, parts procured and systems assembled, a conceptual model is usually built first to allow us to analyze the feasibility of such a system and assess the fundamental characteristics of system performance. Data collected from conceptual models helps designers to make the necessary adjustments; it is usually cheaper and quicker to modify a conceptual system than a physical system. Conceptual systems design is a very critical step for systems engineering design, as most of the systems analysis occurs at this stage and most systems specifications will be determined in the conceptual model. From a conceptual model, a physical model can be easily derived. Conceptual design will be covered in Chapter 2.

1.2.4 Open Systems versus Closed Systems

Systems can be classified based on their interaction with the environment. Open systems exchange information, matter, or energy with their surrounding environments, while closed systems do not exchange such things. In a very strict sense, there is no absolutely closed system existing in the universe, while any system can be more or less considered an open system. Generally speaking, systems are thought of as closed systems when the exchange of matter, information, or energy can be ignored. In physics, closed systems are further classified as closed systems or isolated systems. An isolated system has no exchange of anything with the outside environment, but when only energy is exchanged, the system is closed but not isolated. An isolated system is an ideal closed system, which is practically nonexistent, but closed systems can be found in our daily life, such as a sealed container of water or gas. In thermodynamics, the laws of thermodynamics requires the system to be classified exactly as isolated, closed, or open; thus, system energy and entropy (the amount of energy used for work) can be determined. Open or closed systems concepts also apply in engineering systems and social systems, with different components: large numbers of biological objects or groups that interact with each other, rather than physical particles or objects. In systems engineering, a system is considered open—especially when one takes a life cycle perspective—if the system interacts with the environment and its impact on the environment is one of competitive advantage that system design should include in the early development stages.

Besides the four types of classification, systems can also be classified into other different categories. For example, in system simulation, we often distinguish between continuous systems and discrete systems; in the business and management field, there are manufacturing systems and services systems. The classification is really dependent on the specific system we are studying and the objective of the study.

System science and systems engineering have developed along with the development of technology. System concepts and systems thinking have become more critical in industry, especially after the industrial revolution;

manufacturing has been transformed from human labor-based, small-scale workshops to large factories using complex machinery systems. The transition to the machine age enabled many new machines to be developed; machines gradually replaced human labor in many areas, freed humans from repetitive and dangerous work, increased work efficiency, and in turn made more advanced machines to achieve more complex functions and tasks. After many decades of machine age development, technology has served and replaced almost every aspect of our everyday life, including the manufacturing sector. Now, we can build a new machine without worrying about the details or the fundamental parts of the machines, as we can find the parts to assemble them quickly. The focus has now shifted from the machine age to the system age. The new problem is not to make a single device work; it is how to use the available technology to build something more complex, more powerful, more user friendly, and more efficient. When we begin building systems, systems thinking becomes more important than at any time previously. We need to have the big picture first, just as when we draw a large picture, we compose and sketch the system structure first before we put in any details and colors. A good design process will ultimately determine the success of the system, and the philosophy of the big picture, carried out by a structured process, is called systems engineering. In the next section, we will define systems engineering, review its fundamental features as well as its historical perspective, and introduce the profession of systems engineering.

1.3 Systems Engineering

Before we get into systems engineering, let us first talk about engineering. Engineering, according to the American Board for Engineering and Technology (ABET n.d.), is

the profession in which a knowledge of the mathematical and natural sciences gained by study, experience, and practice is applied with judgment to develop ways to utilize, economically, the materials and forces of nature for the benefit of mankind.

It is the application of what we discover from science in man-made systems. Systems engineering, similarly, applies the knowledge, theories, models, and methods of systems sciences, based on the philosophy of systems thinking, to guide in designing man-made systems. The International Council on Systems Engineering (INCOSE 2012) defines systems engineering as follows:

Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining

customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem: operation, performance, test, manufacturing, cost and schedule, training and support and disposal. 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.

In all the systems engineering books available, there are some variations on the definition of systems engineering, but regardless of what definition is being used, systems engineering has the following common characteristics:

1. It is an applied science. Systems engineering applies scientific discoveries and mathematics to make a specific system work. Note we say specific here; that is because systems engineering does not really address the creation of general scientific theory, nor does it work for all systems. Its results only work for a specific system, not every system in general.
2. Systems engineering is concerned with the big picture of the system: it is top-down design processing, as opposed to starting with detailed bottom-up processing. In systems engineering, a system starts with a need that comes from the users/stakeholders of the system. This need is translated into concepts and is expressed in the format of systems *requirements*. Systems engineering design is driven and guided by requirements, which follows a structured iterative design process that constantly involves “analysis-synthesis-evaluation.” Through this process, the details of the systems are gradually evolved into tangible, visible physical systems. The systems engineering process is a translation process, transforming users’ needs into a working system through guided procedures and proper use of models and analytical methods. In this process, critical thinking and rigorous reasoning (top-down thinking) as well as creative thinking are both important for the success of the system. In turn, systems engineers are required to have a broader scope of knowledge and skills, compared to other traditional engineers; as we will see later, systems engineering is a general design philosophy, which can be applied to all kinds of complex systems design, including aerospace, aeronautics, manufacturing, software, transportation, and so on, to name a few, with special tailoring to the system being designed. A systems engineer not only needs a good understanding of the nature of the system, understanding of the domain knowledge of that specific system, knowledge of how the system works—at least at a high

level—but also have the ability to coordinate with different engineers in the design team. Thus, systems engineers need to have both knowledge of big-picture system thinking and good communication skills. A good systems engineering team is often the core determinant of a system's success.

3. Systems engineering is a multidisciplinary field. As we have discussed the systems engineering process, we have mentioned that systems engineering is multidisciplinary in nature; it is essentially team work. Arthur David Hall III (1969) proposed a three-dimensional morphology of systems engineering: time, logic and profession. The time dimension defines the major design phases and milestones of the system development processes; the logic dimension defines the logical models and steps through which a system would evolve, similar to the system's life cycle status, including problem definition, design, synthesis, decision making, optimization, planning and actions, and so on. The third dimension, profession, describes the different professions that are involved in systems engineering work. Systems engineering professions include management, designers and engineers, graphical artists, and supporting technicians. Compared to several decades ago, at the present time the extent of this range of professions has expanded a great deal, due the scope and complexity of systems we are designing, which have arrived at a different level. With the advanced development of current technology in computing and the Internet, and the fast-growing trend of globalization and supply chain management, there are remarkable increases in the number and scope of professions. The current disciplines and professions involved can be summarized in four different categories:
 - a. The art and science domain: This is the fundamental scientific knowledge pertaining to systems concepts and system analysis methods and models, including applied mathematics (operations research), human-related science (psychology, physiology), graphical design, architecture, environmental science, natural science, biology, and so on. The main mathematical model used in systems engineering is operations research, an applied field in mathematics.
 - b. The engineering domain: Engineering teams are a core operational part of any systems design. Depending on different types of systems, the relevant engineering disciplines are included as a part of the systems design team. Moreover, in terms of system-level planning, scheduling and optimization, industrial engineering, human factors engineering, and management science are among the most relevant fields for systems engineering.

- c. The management domain: In the current globalized environment, any business faces competition from all over the world. Utilizing the global supply chain is an essential part of core competitiveness for any business, translating to systems designers making management a more important role in the design and distribution process. For an effective design, management must not only provide a framework to manage and control the design team and its efforts, but also, more importantly, build a culture and management style that encourages and motivates employees, reinforcing user loyalty throughout the system life cycle and design process. Almost all management functions play a role in the design, including human resources, accounting, finance, and marketing. These functions provide an external liaison for the systems design teams and make the internal operations more efficient as well.
- d. Supporting roles: Besides all the key players in systems design, as mentioned above, there are also personnel that support the design efforts, sometimes indirectly, such as the information technology technicians, legal department specialists, test and equipment technicians, and others. These support functions make sure the daily design activities are carried out smoothly, which is also essential for any design team.

The main professional association for systems engineering is the International Council on Systems Engineering (INCOSE). INCOSE is a non-profit organization dedicated to the field of systems engineering. Founded in 1990, INCOSE has over 9000 members (as of December 2013) all over the world, representing a wide range of expertise and backgrounds from industry and academia. The main mission of INCOSE is to “share, promote and advance the best of systems engineering from across the globe for the benefit of humanity and the planet” (www.incose.org). INCOSE offers professional training and certification in systems engineering; for more information, refer to www.incose.org.

1.4 Brief History of Systems Engineering

Systems engineering is a relatively young field, compared to other engineering disciplines. However, its fundamental concept, system science, can be traced back to ancient literature as early as the eighteenth century, when science knowledge exploded, especially in the natural and physical sciences. The creation of structure in different areas of scientific knowledge set in motion the development of system science. Since then, the fundamental concepts of system science have been presented in many scientific disciplines.

1.4.1 From Reductionism to System Thinking

There are two major milestones in the development of system science that led directly to the growth of the field of systems engineering: the machine age and the system age. They are based on different views of systems. The machine age is largely based on reductionism. Reductionism is a philosophical view of science. It is based on the principle of causality, that is, a system behavior can be explained completely by its fundamental elements or components; it holds the premise that the components of a system are the same when examined separately. Since the early nineteenth century, with the development of new technology, devices and tools, such as the steam engine, the industrial revolution has marked a turning point for modern manufacturing and production. More and more advanced machinery has been developed to replace human power; machines have been developed to achieve more sophisticated functionality. The main principle for designing these machine systems are cause-effect control systems, which are the main characteristic of the machine age. The focus was on designing the control mechanism for the machine; and by integrating different components together, a bigger system can be built. In the machine age, the design of systems largely depends on the reductionism principle, following a bottom-up process. As the technology for individual machines becomes more developed and system development moves toward to a more complex level, reductionism has started to show its limitations when dealing with increasing levels of complexity.

Using reductionism, design methods could fail, especially when applied to complex phenomena such as human society, biological sciences, behavioral sciences, and managerial sciences, as these systems are difficult to examine as isolated entities, and the system is not simply the sum of its elements in a linear manner. The systems approach, however, is more holistic, and follows the concept of expansionism. A system approach argues that even if every part is performing well for its objective, the total system might not be performing well for the system objective if the parts are imperfectly organized (Parnell et al. 2008).

System design moved from the lower level of the machine age to the system age. In the system age, systems approaches look at systems as an integrated whole unit, with the composed components interacting together to serve the systems' purposes, often in a nonlinear manner. The unique characteristic of the system age is the design of systems from a top-down approach; instead of looking at individual components first, system thinking starts with the system as a whole at the beginning, by looking at the big picture of the system, identifying the objectives (requirements) of the system, and having the objective to direct the design of the system. After decades of continuous evolution, systems engineering has become the standard approach for complex system design. In the following section, major historical events are described to briefly show the evolution of systems engineering.

1.4.2 Early Practices

The first recognizable use of the systems approach was in the telephone industry in the 1920s and 1930s. It was Bell Telephone Laboratories who first used the term “systems engineering” in the 1940s. The concepts of systems engineering within Bell Laboratories traces to the early 1900s and describes major applications of systems engineering during World War II. At that time, a group of U.S. and British scientists from various disciplines tried to resolve the problem of achieving the optimal military strategies and actions using limited resources. These practices directly led to the birth of an applied mathematics, known as *operations research*, now a major applied mathematical methodology used in systems engineering. The application of operations research proved substantial through its successful application to military operations. Around the same time, the National Defense Research Committee established a Systems Committee with Bell Laboratories support to guide a project called C-79, the first task of which was to improve the communication system of the Air Warning Service.

In 1946, the Research and Development (RAND) Corporation was founded by the United States Air Force; the systems analysis applied within RAND has been considered an important part of systems engineering development. It was the first to propose and utilize the “system analysis” approach and demonstrate its significance. During this time, the first formal attempt in history to teach systems engineering started at the Massachusetts Institute of Technology (MIT) in 1950 by G.W. Gilman, the director of systems engineering at Bell Laboratories. With the wide application of operations research and control theory, together with the development of digital computers, the scientific foundation for systems engineering has been established.

1.4.3 Government Role

Government has played an important historical role in promoting systems engineering, since most of the early complex systems were requested by government. With more demand for and experience in developing large complex systems, the need to identify and manipulate the properties of a system as a whole—which in complex engineering projects may differ greatly from the sum of the parts’ properties—motivated agencies such as the Department of Defense (DoD) and National Aeronautics and Space Administration (NASA) to pay more attention to applying systems engineering discipline. Since the late 1940s and early 1950s, the DoD has applied a systems engineering approach to the development of missiles and missile-defense systems. This undertaking was recognized as the Intercontinental Ballistic Missile (ICBM) Program, later the Teapot Committee, and it served as one of the key historical foundations of systems engineering. The ICBM was an idea given by Bernard Schriever to the Teapot Committee in 1954, and it ultimately changed the organizing principles of managing a systems development contract.

From this point on, “there would be a System Engineering contractor staffed by ‘unusually competent’ scientists and engineers to direct the technical and management control over all elements of the program” (Hallam 2001). The Teapot Committee helped establish systems engineering as a discipline by “creating an organization dedicated to the scheduling and coordinating of activities for subcontractor R&D, test, integration, assembly, and operations” (Hallam 2001). The coordination required for the system approach to the project was further driven by schedule demands containing concurrent development of designs, subsystems, manufacturing, and so on. It was then, through trial and error, that quantitative tools and methodologies were developed. These very tools formed the basis for the interdisciplinary trade studies and decision aids that enrich the field of systems engineering.

By the 1960s, degree programs in the discipline of systems engineering became widely recognized across U.S. universities and many systems engineering-related techniques were developed within more academic research and industry applications. For example, in 1958 the Program Evaluation and Review Technique (PERT) was developed by the United States Navy Special Projects Office for the Polaris missile system. PERT is a scheduling method designed to plan a manufacturing project by employing a network of interrelated activities, coordinating optimum cost and time criteria. PERT emphasizes the relationship between the time each activity takes, the costs associated with each phase, and the resulting time and cost for the anticipated completion of the entire project. Since existing large-scale planning was inadequate at the time, the Navy teamed up with the Lockheed Aircraft Corporation and the management consulting firm of Booz, Allen, and Hamilton, working in large-scope systems engineering efforts. Traditional techniques such as line of balance, Gantt charts, and other methods were found limited in dealing with variability, and PERT evolved as a means to deal with the varied time periods needed to finish the critical activities of an overall project.

By the mid-1960s, in response to many contractors expressing a need for greater latitude in applying alternative systems engineering techniques, several criteria-oriented military standards were issued, including MIL-STD-499A. Army FM 770-78 and MIL-STD-499A formed the foundation for the current application of systems engineering concepts and requirements in military development programs. It was in the late 1950s and 1960s that the emergence of engineering discipline “specialists” on most development programs occurred.

In 1962, Arthur Hall published his first book on systems engineering: *A Methodology for Systems Engineering*. Hall was an executive at Bell Laboratories and was one of the people who were responsible for the implementation of systems engineering at the company. Hall reasoned that systems engineering was important because products were increasing in complexity, the needs of consumers were expanding, the business markets were growing rapidly, and there was an acute shortage of technically and scientifically trained people handling the complex systems design. Systems engineering was in great

demand in this era. According to Hall, systems engineering is a function with five phases: (1) systems studies or program planning; (2) exploratory planning, including problem definition, selecting objectives, synthesis and analysis, selection, and communication; (3) development planning; (4) studies during development, including system components development and integration and testing of components; and (5) current and emerging engineering, that is, current engineering technology and that which will exist while the system is operational (Buede 2009). Hall's book is considered a major milestone for systems engineering education.

In the 1960s, a third defining moment for systems engineering was seen: the birth of NASA's Apollo program, which lasted from 1961 to 1972. In fact, the Apollo program is probably the most classic example of systems engineering in practice to date. The task of sending humans to the moon was daunting and complicated; it involved breaking the underlying goal into multiple sections or manageable parts that participating agencies and companies could work with and comprehend. These various parts then had to be reintegrated into one whole solution, and as a result, careful attention and management involving extensive testing and verification was necessary. The complex nature of these tasks made systems engineering a suitable tool for designing such systems. It was the principles of systems engineering that resulted in the rigorous system solution which contributed to Apollo's overall success.

In 1972, the International Institute for Applied Systems Analysis (IIASA) was founded, with the intention of applying techniques of systems analysis to solve urban, industrial, and environmental problems that transcended international boundaries. Members of the IIASA discussed hallmarks of the IIASA approach to problem solving, which included interdisciplinary emphasis and the maintenance of credibility with both scientist and decision makers (Hughes, n.d.). The IIASA approach is illustrated in the organization's most successful venture: a transboundary air pollution project using the Regional Acidification Information and Simulation (RAINS) model of the impact of acidification in Europe. With efforts such as these, systems engineering has emerged from the government and military fields to almost every area of the industrial and social, and moved to a more comprehensive and advanced stage: the information age.

1.4.4 Information Age

With the development of hardware and software, computer technology made tremendous advances in the 1980s and more in the 1990s, which marked the beginning of the information age, or computer age/digital age. With fast-growing and innovative technologies such as the internet, humans have achieved an era of most effective, rapid technological growth than at any previous time. More and more complex, technology-driven systems are being designed every day, creating a new technological world for a wide range of users. With the shift from traditional industry to a highly computerized

economy, the degree of complexity involved in the systems has reached a new high level. The increasing level of complexity has put the practice of systems engineering in greater demand. With the blooming of information technology, limited resources dwindling, and a more competitive market, the world has started a whole new chapter, characterized by globalization and the use of the supply chain in the design process. It is hard to believe that, at the current time, any organization or business is not using some kind of global supply chain for their operations. Having an efficient systems integration process is essential for the success of any complex system design. In the information age, or system age, it is hard to find any complex system which does not follow a systems engineering approach. Nowadays, systems engineering has become a well-accepted standard for practice across government and industries internationally.

In 2009, CNN Money ranked systems engineer as no. 1 in the list of the best jobs in the United States, according to CNN.com:

demand is soaring for systems engineers, as what was once a niche job in the aerospace and defense industries becomes commonplace among a diverse and expanding universe of employers, from medical device makers to corporations like Xerox and BMW.

After several decades, systems engineering has now grown into a profession; it is a must-have function and department for most of the large corporations, and the major association for systems engineering, INCOSE, has grown significantly since its formation in 1990. Now, INCOSE has more than 9000 members globally, and is still growing every year. Meanwhile, government and industrial organizations are still facing a significant shortage of systems engineers, especially senior-level expert systems engineers; careers in systems engineering will become more abundant in the near future. That is the reason why we are here, to study this subject.

1.5 Summary

Systems engineering has become one of the most important engineering fields in every aspect of our society. It is based on systems concepts and the philosophy of the big picture first, in order to bring complex systems into being from beginning to end in the most efficient manner. This first chapter reviewed the basic concepts of systems, systems science, systems engineering, and the historical evolution and development of systems engineering. Knowing this basic information about systems will help us understand the origin and need for systems engineering and thus grasp better the ideas and models in systems engineering.

PROBLEMS

1. Define systems. What are the characteristics of systems?
2. Give an example of a system within the systems engineering context. Identify the basic elements involved.
3. What are the system classifications? Briefly discuss these classifications and give examples (other than those examples given in the book).
4. Define science and engineering. What is the difference between science and engineering?
5. What is reductionism and what is expansionism? Briefly discuss the difference between these two terms.
6. Describe the main historical phases of systems engineering development and the main development milestones for each phase.

2

Systems Life Cycle and Design Processes

Let us start the chapter with a “rocks and bucket” story. There is a bucket, some big rocks just enough to fill it, some small stones, some sand, and water. Let us first fill the bucket with big rocks. Do we still have space? Yes, so we put the small stones around the big rocks in the bucket. Now is it full? We still have some room for some sand; after putting the sand in, is it full? No, we can still put water in the bucket: now the bucket is full. Now ask yourself: What can you see from this story? Someone might say that you can always have room for smaller things. While this might be true to some extent, for us systems engineers, the point we get from this story is that for us to fit all the things (big rocks, small stones, sand, and water) into the bucket, we have to follow the order in the story. If we put the small things into the bucket first, in the end we will realize that there is *no way* we can fit all the big rocks into the bucket. This order is analogical to the fundamental philosophy guiding the system engineering process: top-down processing.

The whole idea of systems engineering is the implementation of a top-down process; that is, starting from a problem or a need, it gradually evolves from a design concept to a tangible system that can be used. The key issue in a top-down systems engineering design process is a *big-picture* perspective; that is, designing the system from a life cycle perspective. Nothing lasts forever; everything has a life span. Just like anything else in the world, an engineered system also has a life cycle. Utilizing the big picture means that we have to consider everything involved in the system life cycle from the very beginning of the system design, doing things right in the beginning, even on issues related to the system’s retirement. They need to be considered in the design phase, as design decisions in the early stages can have a significant impact on the system’s performance later, even on the system’s retirement. One of the most important factors in systems design is change or violation of design needs. It has been found that a large percentage of costs incurred in systems design is due to changes in the later design stages. Although, for any system, it is inevitable that change will occur, since unexpected and random events happen all the time, systems design must have the ability to accommodate such an environment; however, controlling and minimizing those changes can significantly reduce the complexity and cost of the whole design process, as the later the change occurs, the more expensive and difficult to implement it becomes. Having a big picture at the beginning of the design process helps us to keep the design changes at a minimum level. The second issue for systems engineering is that it is process oriented; in other

words, the process determines the success of the system. Every system starts with a need and it gradually evolves to its final forms through a structured, well-defined process. There is no guarantee that a good process will lead to a good product, since many unexpected issues will occur during the process; however, a structured process will provide the greatest probability of success for bringing the system into being within its cost and time limits. After many decades of practice and implementation, the systems engineering process has been proven to be the most effective and structured approach for designing systems successfully, especially for large complex systems, and it is believed that it will still be the philosophy used for complex system design in the foreseeable future. In this chapter, we will

- Provide an introduction to systems' life cycles and describe the major forms of systems at different stages of system life cycles.
- Describe the elements that are involved in the system design process, including system need, requirements, functions, components, prototypes, models, and their relationships with regard to different phases in systems' life cycles.
- Describe the system engineering process and define the main objectives for each of the phases in the design processes, including the main design activities, major milestone baselines, and product for each of the phases, and the design evaluation approaches applied in each of the phases.
- Review the most common models for the systems design process, including the waterfall model, vee model, and spiral model, and their unique features for different types of systems.

2.1 System Life Cycle

"Life cycle," according to *Webster's Dictionary*, refers to "a series of stages through which something (as an individual, culture, or manufactured product) passes during its lifetime." Almost everything has a life; even the sun on which our planet Earth relies has a life of approximately 14 billion years, starting from a large cloud of dust and gas 5 billion years ago, to become a giant red star in another 5 billion years, and eventually a black dwarf approaching the end of its life.

The life cycle of an engineering system is a sequence of stages/phases in the life of the system. It is the life span or history of the system, from the need to create the system to the point that the system is retired and removed from service. There are some variations among different people regarding the naming of the systems life cycle; for example, Clark et al. (1986) defined

systems life cycles as stages of “planning, definition, design, integration acceptance, delivery product”; Blanchard and Fabrycky (2006) used the terms need identification, conceptual design, preliminary design, detailed design, production and construction, utilization and support, and systems phase-out and disposal. Some of the life cycle definitions, such as the ones from Blanchard and Fabrycky, are more like a design process, expressed from the designer’s perspective. Here I adopt the one from Chapanis (1996), as it concentrates on the system’s status and format at different life cycle stages, rather than focusing solely on the design activities. Before we get into the life cycle description in detail, there are two fundamental characteristics of the systems life cycle that need to be brought to the reader’s attention. Firstly, any system starts from a need; this need determines the system’s concepts. From the concepts, system functions are derived, decomposed, and allocated to the system’s components—hardware or software—all the way to its final configuration: developed, constructed, and delivered. All the different stages of the systems life cycle and different forms of systems are driven by the initial needs. Secondly, from the first characteristics just mentioned, one can easily see that the system evolves from a general form to specific forms. This evolvement is a top-down process, starting with the big picture—the need for the design of the system—with further details added in the various life cycle phases. In the following section, we will describe the major phases in the system life cycle; these life cycle phases summarize a common form for most systems. Despite the fact that different systems have different variations of these forms, the fundamental phases should be similar. Generally speaking, the system life cycle consists of seven major phases: (1) the operational need for the system; (2) system concept; (3) system concept exploration and validation; (4) engineering model development; (5) systems production, deployment and distribution; (6) systems operation and maintenance; and, finally, (7) system phase-out and retirement. We will briefly discuss, in each of the system life cycle phases, what the basic format of the system is, what major questions are being addressed, what information it will include, and what major system outcome is expected for these life cycle phases.

2.1.1 Operational Need

A system starts with a need; a need gives the reason for such a system—that is, *why* does one need such a system and *what* is needed—at a very high level. Needs come from customers. There are two basic types of customers involved in systems engineering (these are not exactly the same as the system users; we will talk about system user classes in greater detail in Section 2.2.3). One type is the government agency; for example, the Federal Aviation Administration (FAA) may want a new flight management system (FMS) to be developed, based on the electric flight bag (EFB) devices to integrate four-dimensional (4-D, spatial plus temporal) flight data into the cockpit. Government agencies such as the FAA and Department of Defense (DoD)

play an important role in prompting systems engineering, as most of the systems they need are large and complex, and often have a tight budget and schedule. These government systems usually have more strictly defined structures and specifications; for example, for the DoD, a commonly used structure is Department of Defense Architecture Framework (DoDAF): this is a foundational framework for developing and describing systems architecture that ensures a common denominator for communicating, understanding, comparing, and integrating architectures across different organizations for DoD-related systems. DoDAF establishes data element definitions, terminologies, rules, relationships, and a baseline for consistent development of military systems architectures.

The other type of customers are nongovernment customers, either commercial or nonprofit related. Systems or products needed for this type of customer vary a great deal in terms of scope and complexity, and compared to government customers, their needs are usually less structured. For example, a marketing survey found out that U.S. customers need a new type of hybrid vehicle that uses environmentally friendly fuel, and in addition, need to be able to use household electricity to charge the vehicle, due to the rising price of regular gasoline. In areas that lack electricity, there is a need for a solar-powered portable water purification system, to provide drinking water for a certain number of people daily.

Systems needs originate from different sources; for government systems, the needs are usually generated from national strategic decision-making, and documented and distributed by a request for proposal (RFP). RFP is a formal invitation document, distributed to the prospective firms or academic organizations, to describe the needs for the system to be developed, and inviting the qualified organization to propose a systems development plan. Based on the RFP, potential developers will respond with a structured proposal, describing the system in a more technical and operational form, and propose a solution and plan for the system. The successful bidders will be required to develop a detailed work plan—namely, a statement of work (SOW)—to illustrate the technical and managerial aspects of systems development, including the personnel qualifications, efforts, and the major timeline and budget. The detailed information included in the RFP and SOW will be explained in Chapter 3.

For nongovernment customers, the need for a new system or product originates from organizational strategic planning. Strategic planning for a new system or product emerges from a number of sources, including marketing surveys and customer feedback/complaints, changing environments, new technologies and new resources, depletion of old resources, and so on. Unlike government projects, in which a strict documentation format has to be followed, in nongovernment projects, the operational need is developed internally and does not usually have a unified format which must be adhered to. Regardless of what type of systems are developed, one cannot depend solely on customers' original views as the only source, as these are

often vague, incomplete, and too general. That is why one needs to have an operational need. Information from a variety of sources needs to be compiled and translated into one document, describing the system's intended purpose and functions. The following example illustrates typical operational needs:

1. Introduction: This section gives the background and scope of the systems to be designed/requested.
2. Missions: This section defines the highest level of mission that this design tries to accomplish. Missions may be broken down into specific phases and time periods if required.
3. Technical objectives: This section defines the major functions required for the system to be designed. Major function performance parameters are included for these functions. These functions/parameters may be accompanied by technical performance measures.
4. Constraints: This defines the time/cost restrictions on the project and the major milestones and deliverables required.

A sample technical need document can be seen in Table 2.1.

2.1.2 System Concept

After the operational needs are identified, an integrated systems concept will be developed. This concept is illustrated by a conceptual model. A model is the abstract representation of the real-world object; it focuses on the factors most relevant to the system needs, conveying the important relationships between these factors, and ignores nonrelevant factors. For large complex systems, using appropriate models is of the utmost importance, as there are usually hundreds or thousands of factors involved in the design. For the most part, this book is about the models used in systems

TABLE 2.1

Sample Design Needs for a Small Aircraft

Parameter	Description
Range	200 statute miles, with 30 min reserve, day VFR at ≥ 4000 feet MSL over nonmountainous, sparsely populated coastal terrain
Efficiency	≥ 200 passenger-MPGe energy equivalency
Speed	≥ 100 mph average on each of two 200 mile flights
Minimum speed	≤ 55 mph in level flight without stall, power and flaps allowed
Takeoff distance	≤ 2000 feet from brake release to clear a 50 foot obstacle
Community noise	≤ 80 dBA at full power takeoff, measured 240 feet sideways to takeoff brake release

engineering. We will explore different types of systems engineering models in Section II. The philosophy of systems engineering can be considered to be model based, as models are an absolute necessity in all aspects of systems engineering design. The conceptual model is the first model that the system evolves from its need. In the conceptual model, systems needs are organized as *requirements*. Requirements are formal documents that define the system's intended purposes. From the requirements, operational concepts will then be developed. These depict a complete picture of the systems operations. Operational concepts are commonly written in narrative format; sometimes they are also called operational scenarios. Scenarios tell a complete story of the system's intended activities, if designed and developed, and how it will be utilized by its operators/users. The following example shows a typical scenario for using an automated teller machine (ATM):

The customer (including walk-up and drive-through customers; they may also be visually and hearing impaired) requests service by pressing the start key or by inserting their debit card. They receive feedback from the ATM that their request was accepted. The ATM system reads the card and requests the PIN. The customer inputs their PIN and the system processes it; if the PIN is correct, the ATM proceeds to the next selection menu. If the PIN is not correct, the ATM provides feedback and goes back to request the PIN again. If this process is repeated three times, then the ATM blocks the card and provides feedback.

To better explain the system scenarios, some other format of the operational profile can also be developed to accompany them, to give a more comprehensive picture of the system concepts. This profile includes a graphical representation of the systems operations spatial profile, a timeline of the operations temporal profile, a chart of user/operator characteristics, and an interaction diagram to illustrate the exchange of information, materials, and energy flow between users, systems, and the environment. We will describe this in depth in Chapter 3, which is about systems requirement analysis.

As an immediate result of systems concepts development via the methods mentioned above, systems evolve to a conceptual functional model. System *functions* define the purpose of the system and illustrate *what* actions the system would perform to accomplish its purposes. The concepts of functions are developed to give a complete picture of the system's intended actions, hierarchically and operationally. It defines what the system should do to accomplish its requirements and to fulfill customer needs. Functional analysis is one of the most important systems analysis methods, as it serves as the first and most critical bridge between customer needs and the system's technical specification. Functional analysis will be covered in great detail in the systems engineering process sections and in Chapter 4.

2.1.3 Systems Concept Exploration and Validation

Systems concepts, represented by system functional models, need to be further explored in terms of their validity to the system requirements. System concepts are explored and validated by translating the conceptual functional model to the allocated *components* model. A component allocation model is built to verify that, firstly, the conceptual model is feasible to be implemented by systems components, including hardware, software, and human operators. It is a two-way process, in the sense that the conceptual model needs to be adjusted to make the physical allocation feasible, taking into consideration the availability of the current and emerging technology; on the other hand, the functional model serves as the basis for the selection of the components, as there are many candidates for these, with a large variety of suppliers, manufacturers, and different models involved. The decision-making process for the translation from conceptual model to physical models is iterative and requires a careful consideration of all the requirements and their parameters. That brings us to the second verification in this phase, which is that systems requirements are verified by the physical models. You might have seen, on several occasions, that we have used the words *verification* and *validation*; you might wonder whether these two words are the same? If not, what is the difference? In the systems engineering context, “verification” and “validation” are related to each other but do not convey exactly the same meaning. The verification process is to make sure the systems requirements are being met at different phases in the system life cycle; that is, the systems requirements are translated, carried over, implemented, and materialized throughout the system life cycle. Systems requirements are traced through this translation, and the design models are verified against the requirements parameters by using reviews and tests at different stages; sometimes, users are also involved in the tests and reviews. Validation is similar to system demonstration or evaluation: it is to make sure the design outcome is what users want, and that we are designing the correct systems. So, briefly summarizing the two terms, verification means doing things right, following the systems engineering process and structure, while validation is to make sure we are making the right thing, providing the correct outcome for the users’ needs. These two activities go hand in hand, and are closely related to each other; without a good verification plan and process, there will be no valid system to be designed.

In the systems exploration and validation phase, systems evolve from the conceptual model to the physical allocation model, represented by the system allocation baseline (Type B system specification). The allocation baseline answers the question of *who* will perform *what* function at what cost. One has to keep in mind that the process is iterative; the allocation models of the systems are reviewed, and usually systems users are involved in this review process.

The allocation model usually includes the following items:

- Systems requirements for subsystems and components, refined and quantitative technical performance measures for functions, including the input, output, and constraining factors for each function
- Functional package and assembly lists, including systems components lists and preliminary configurations for the components
- User systems interaction requirements
- Design data, validation and evaluation documentation, including the faulty tree analysis (FTA) and failure mode and effect criticality analysis (FMECA), human factors tasks analysis, and computer-aided design data

2.1.4 Engineering Model Development

Based on validated system concepts, functional models, and allocation baseline, the actual components are procured and assembled. This is the point at which all the components are integrated together and, for the first time, the system is in its intended form, not only functionally but also in its physical configuration. To obtain a complete and detailed system configuration, the systems specification is translated from *what* (functions) and *who* (allocations) to *how* and *how much*. Details of systems components are specified, including the systems component/assembly specifications, what materials are involved, and how to assemble them together. The system evolves from the functional allocation baseline (Type B specification) to the product baseline (Type C specification), process baseline (Type D specification), and material baseline (Type E specification). There are various types of documents involved in the engineering model, based on different types of systems. Generally speaking, the model usually consists of the part list, material lists, blueprints, computer-aided design data (such as AutoCAD and CATIA design drawings), and specifications for the interfaces between components. Since this is the first time that all the components are integrated in their actual design forms, the system will be tested iteratively to verify that the derived physical models meet the system requirements. These tests, usually conducted formally with users and stakeholders involved, are intended to identify the last-minute problems before the system is produced, distributed, and deployed. Necessary changes are made to the models if any mismatches between the physical models and system requirements are found, trade-off analyses are carried out to balance out different options, and changes are monitored and strictly controlled in order not to cause any major time delay and cost increase. Minimizing changes is one of the most important objectives for systems engineering. But, due to the complexity of system design, changes are often inevitable. Systems engineering tries to identify and address the changes as early as possible, which is why traceability plays an important role in the life cycle. Different

system formats are interconnected, and the evolution of different models from one version to another is supposed to be requirement driven, to minimize any deviation from the right track. We will talk about change management in the detailed design phase later in this chapter. Here, it is important to point out that managing changes efficiently is an essential aspect of the development of physical system models.

2.1.5 System Production, Distribution, and Deployment

After the physical system model has been built, tested, and finalized, the system will move to the next life cycle phase, which is full-scale production, distribution, and deployment. The system is in its final format, and is transferred to the production assembly line to start formal mass production. At this stage, the system is produced, possibly in multiple copies. All the components are specified, either produced separately if they are specially designed for the system, or procured if using standard commercial off-the-shelf (COTS) items. These components are delivered to the production site, assembled, and then distributed via retail outlets to customers. A final production of the system may involve hundreds or thousands of suppliers, depending on the level of complexity of the system. The supply chain plays an important role at this stage, as greater cost may be incurred in distribution and deployment than in the production/assembly itself. Much of the system's assembly work may be outsourced internationally; for example, to take advantage of cheap labor and materials locally. For instance, the cost of labor and materials of many U.S.-designed systems in third-world countries, such as China, is less than 10% of the total cost of the system. As a matter of fact, the majority of the cost is incurred in system distribution and deployment, in addition to the cost of research and development. The system has now evolved from its models to the final realization of its designed format, together with its supplemental materials including manuals and training services; it is delivered to customers and installed for operations.

2.1.6 System Operations and Maintenance

After the system has been produced, assembled, and delivered to the customer, it will be in fully operational mode for the period of time designed, providing functions to users and operators. Users and operators are usually the same group of people; the difference between these two lies in the complexity of the system itself. If the system is fairly simple in nature, the term *users*, that is, the user–equipment combination, is employed; if the system is very large and complex, the term *operators* is used instead. System users and operators essentially represent the same type of user class; namely, the end customer of the system. There are also other classes of systems user; systems maintainer is one of the classes. Although highly reliable systems are desired, systems do fail sometimes, requiring

maintenance activities to be carried out. At this stage of the system life cycle, the emphasis is on customer service and support of the system; since the design of the system has been finalized, no further design changes are possible. However, follow-up tests and evaluations of systems are still necessary to identify any problems within operations and maintenance; these test results, together with feedback from customers, serve as a guide for any engineering changes that may be made for the subsequent version or next generation of the system. Emerging problems are addressed and fixed immediately, especially those pertaining to user safety and hazards. Faulty systems are fixed, recalled, or replaced to minimize the impact from these issues. At this stage of the system life cycle, operation, maintenance, and evaluation efforts are carried out continuously, until the system is ready to be retired.

2.1.7 System Phase-Out and Retirement

There are many reasons for system retirement; the majority of these are incompatibility with emerging technology, discontinuation of supply of materials, changes to legislation and regulations, new trends in customer demand, and so on. At the system retirement stage, the system is often characterized by a reduction in the number of customers, increasing numbers of system problems, high costs, and difficulty of maintenance. System functions are terminated and the system is disassembled to the level of its components; the system is disposed of. The natural resources from which the system is built are limited, so it is unlikely that the system will be completely discarded and wasted; it is desirable to retrieve materials from retired systems as far as possible, to save the cost of the materials, and more importantly, to reduce the amount of waste to the environment. Sustainable development or *green engineering* is one of the most important types of development, depending on the different kinds of components. There may be different end uses for the components: reuse, remanufacturing, or recycling. *Reuse* is the highest level at which a system is preserved—usually a nonfaulty system—to keep it at a degraded function level to prolong its life cycle (one can think of this as a semiretirement phase). For example, older computer systems may not be fast enough for laboratory scientific computation purposes, but since they are still functional, they may be donated to charity for educational purposes in schools and offices. *Remanufacturing* (also known as closed-loop recycling, or sometimes called refurbishing) implies a series of activities to put a retired system back into use in its complete form, by repairing or replacing faulty parts, to become operational again. *Recycling* is a process that retrieves useful raw materials, to reduce waste and the cost of procuring similar fresh new materials. It is believed to benefit the environment by reducing pollution and saving the energy consumption of obtaining new materials. Although a common practice for a long time, it did not catch people's attention until the twentieth century; as an outcome of the industrial revolution, productivity

has increased tremendously, as has demand for materials. With more human-made systems, the by-products, waste, and pollution from the production process has influenced our natural environment, which has caused many problems for human health and quality of life. Sustainable development and green engineering of systems have become more and more important for assessing their effectiveness. With more customer awareness and globalization of system development, green engineering has become an essential part of system competitiveness. The life cycle consideration of system design enables designers to take systems retirement into account in the early phases of the design process—assembling it in a way which will simplify its disassembly—to facilitate the system being phased out and retired, to impact the environment at a minimum level, and meanwhile save internal costs and energy.

Where the system is retired only partially, its materials will be recycled or reused for the next generation of the system and some of the concepts will be carried over to the next generation, as part of the requirements for the new system; at that time another life cycle will begin.

2.2 Systems Engineering Processes

2.2.1 Definition of Systems Engineering Process

Parallel to the system life cycle, there are a series of activities that bring systems into being; this series of activities is called systems engineering processes, or more specifically, system design processes. According to INCOSE.org, the systems engineering process is comprised of seven typical tasks: “State the problem, Investigate alternatives, Model the system, Integrate, Launch the system, Assess performance, and Re-evaluate. These functions can be summarized with the acronym SIMILAR: State, Investigate, Model, Integrate, Launch, Assess and Re-evaluate” or SIMILAR.

Chapanis (1996) also defined systems design as a seven-step process in his process model: “Requirement Analysis, Systems Design, Hardware and Software Design, Hardware and Software Development, Systems Integration, Installation/Transition & Training, and Maintenance & Operation”; the three strands of development hardware, software, and users are woven together into one integrated system. Blanchard and Fabrycky (2006) defined the systems design and life cycle as a process with two major steps, the acquisition phase and the utilization phase; the acquisition phase consists of conceptual design, preliminary design, detailed design and development, and production and construction; the utilization phase consists of product use, phase-out, and disposal. There are many other variations of the processes. Despite the different names for