SERVICE SYSTEMS AND INNOVATIONS IN BUSINESS AND SOCIETY COLLECTION

Jim Spohrer and Haluk Demirkan, *Editors*

Modeling Service Systems

Ralph D. Badinelli

BEP BUSINESS EXPERT PRESS

Modeling Service Systems

Modeling Service Systems

Ralph D. Badinelli

BEP BUSINESS EXPERT PRESS

Modeling Service Systems

First published in 2016 by
Business Expert Press, LLC
222 East 46th Street, New York, NY 10017
www.businessexpertpress.com

ISBN-13: 978-1-63157-023-0 (paperback)
ISBN-13: 978-1-63157-024-7 (e-book)

Business Expert Press Service Systems and Innovations in Business and Society Collection

Collection ISSN: 2326-2664 (print)
Collection ISSN: 2326-2699 (electronic)

Cover and interior design by Exeter Premedia Services Private Ltd., Chennai, India

First edition: 2016

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America.

Abstract

This book invites the reader on a journey of discovery of service systems. From a Service-Dominant-Logic perspective, such systems are the building blocks of all economic activity, and innovation of new service systems holds the promise of a new industrial revolution. Users navigating websites, customers interacting with intelligent mobile retail applications, patients interpreting advice from health-care professionals and other sources, students interacting with teachers and learning materials, city dwellers invoking smart service applications for transportation routing, and the unlimited variations of smart service systems that will be enabled by the Internet of Things and other technologies provide ample evidence of the need for service innovation. Fundamentally human centered and cocreative, these services must engage actors in personalized journeys directed by their decisions. Hence, understanding the performance of service systems and designing better service systems require an understanding of how actors or their agents make decisions and how service systems should enable and respond to these decisions. Service science is the study of such systems and decisions.

This book presents an overview of the foundational constructs of service science and models of cocreative systems, with the aim of enabling the reader to be a service innovator. Consequently, the book's title expresses the purpose of the book in terms of initiating the reader in the action of modeling as opposed serving as a presentation of models for observation. Some readers may possess in-depth knowledge of some aspects of service systems that this text only surveys. That's fine. The value proposition of this book is the opportunity to fill each reader's knowledge gaps and offer a comprehensive, coherent, and introductory overview of service system modeling.

Keywords

cocreation, decision model, modeling, service, service innovation, systems

Contents

Preface

I chose the title of this book to be Modeling Service Systems instead of Models of Service Systems because I consider the book a call to action. In this book, I strive to attract the active reader and to activate the tentative reader. This book is designed for managers, engineers, software developers, system developers, and entrepreneurs in the new service economy who find themselves tasked with building services that are viable and competitive but who have struggled to find appropriate tools for innovating service, have recognized that service does not succumb to the principles and modeling tools of manufacturing or production enterprises, and have been exposed to the advances of the past 10 years in service but don't yet fully understand them.

Researchers in the new field of service science will also find the book useful for reconciling the many diverse viewpoints that have emerged in the service science community. Service scientists, marketing scientists, economists, psychologists, sociologists, operations researchers, and scholars from other disciplines have cocreated essential foundational principles of this science. It is not surprising that such a diverse community engaged in generating disruptive theories and concepts finds itself yearning for a concise and coherent collection of terminology and principles. Perhaps this book can serve to broaden and rationalize the perspectives of researchers and encourage further scientific debate directed toward a unified theory of service. However, the primary purpose of this book is not so lofty. These pages should be most useful to the practitioner.

The book is pragmatic. It is a basic toolkit for designing a service down to the operational level. Through this book, the practitioner can get started in bringing analytical tools to innovating, managing, and evaluating service. Underlying theory is mentioned and referenced, but the focus is on how the theory can be applied.

Why is such a book needed? The service revolution is upon us. Cloud computing, mobile computing, the Internet of Things (IoT), and cognitive computing have leveraged the Internet into a ubiquitous service-providing platform. The app economy is proliferating myriad service components and integrating these components into rapidly configured mashups that offer an overwhelmingly expanding array of services. Our lexicon of service innovation has evolved from smart devices to smart cities to smart countries. It is my belief that achieving the potential of service innovation now mandates a scientifically based, practical methodology with the same force by which the industrial revolution engendered the discipline of industrial engineering 100 years ago.

Can we achieve the same degree of sophistication and precision in service that the past century of research and application has brought to manufacturing? As one who was schooled in the modeling of industrial systems and the power of model-based applications in product supply chains, I am unreservedly optimistic about the prospect of a world of service systems with efficiency in their design, construction, maintenance, and operation that outpaces anything that we have seen in manufacturing. But we cannot achieve this by simple extension of our knowledge base in the manufacturing domain. It is clear that service systems are not special cases of product-based supply chains and manufacturing. On the contrary, everything that 100 years of scientific research in goods-dominant systems has taught us is a special case of the complicated and complex world of service. For this new domain of study, new model constructs and new modeling approaches are needed. We have discovered some of these constructs and approaches, and this book was written with the aim of kick-starting some practical model building and application.

Where does the naïve service innovator start a rational and practical study of service science? For starters, a background in Service Dominant Logic (SDL) is necessary to understand the definition of the word service. Upon first exposure to the theory of SDL, the uninitiated may be surprised to discover that he or she had not known this definition. SDL illuminates the essence of the science of service. The modeling constructs described in this book are all derived from SDL's foundational principals. I recommend to anyone who is new to this science to begin the journey of discovery with the papers by Vargo and Akaka (2009) and Sampson and Froehle (2006) for a clear and concise exposition of this essential foundation.

There are many ground-breaking researchers such as those just identified who have lighted my path in service research. There are too many to list here, but I wish to acknowledge two in particular—fellow board members of the International Society of Service Innovation Professionals (ISSIP), who were responsible for convincing me to write this book. Jim Spohrer of IBM needs no introduction within the service research community as he has been a leading scientist, author, promoter, resource integrator, and seer since the new science of service emerged. I am proud and fortunate to know him as a colleague and a friend. Haluk Demirkan, professor, researcher, and tireless champion of service science continues to be a valued colleague, sounding board, and comrade in service research. I look forward to many more years of fruitful collaboration Jim and Haluk.

Finally, I wish to leave the reader of this Preface with an acknowledgment of a different sort. Several years ago, Jim Spohrer invited me to become a founding board member of the new professional society of ISSIP. Unique in its structure and mission, this society has also become for me an indispensable college of supportive and enlightening coinvestigators of the fascinating discipline of service.

CHAPTER 1

Introduction

Value Proposition

Reading this book is a service. While reading this text, you will be stimulated to think about service and service systems in ways that are new and valuable to you. Through this text, I, the author, will cocreate with you, the reader, added value in your ability to identify, describe, evaluate, design, and manage service systems. Exactly how the information in this book will influence your understanding is a unique and somewhat unpredictable outcome of the process of reading. A more effective, but less efficient, service would consist of a one-to-one dialogue between you and me through which we could customize the integration of our knowledge resources. However, for practical reasons, this text will have to suffice to initiate an education in modeling service systems. I encourage you to contact me if you would like to pursue a deeper understanding of service system models than that which is available in this text.

If you have opened this book, then you must be seeking value in the form of understanding how service is designed, managed, operated, or evaluated. The fact that service has become a ubiquitous function within our economies, professional lives, and personal lives has motivated a burgeoning interest in enhancing the efficiency and effectiveness of service (Ng 2014). As the industrial revolution spawned the 100-year development of modeling manufacturing supply chains through the fields of industrial engineering and operations research, so too will our emerging awareness of service instigate a scientific approach to service. An intelligent approach to any of these aspects of a service enterprise requires a model-based understanding of the system by which service occurs. Hence, the motive for this book is the need for sophisticated and scientific representations of service systems. Accordingly, we will hereinafter refer to you, the reader, as a service modeler.

Some terminology needs to be clarified. We are used to thinking of business enterprises beginning with an invention that is designed by an engineer who applies scientific principles to a practical solution to a problem. The invention is commercialized by business managers who identify, acquire, deploy, and coordinate the resources that are necessary to produce and market the product. Finally, the supply chain for the product is supervised and operated to form a successful enterprise. Does this scenario apply to service? Perhaps not. IBM coined the term Service Science Management and Engineering (SSME) to popularize the company's view of the service enterprises being comprehensive business entities as opposed to offshoots of conventional goods-dominant firms (Hefley and Murphy 2008). I never figured out why management was placed seemingly out of sequence between science and engineering in this acronym, but I do not think it matters, as the SSME acronym has been replaced in the service community with other, more appropriate terms. For reasons that will be explained later, we will use the terms *science*, *innovation, operation*, and *evaluation* to identify the hierarchy of activities that lead to a service enterprise.

Perhaps you don't know whether or not you are interested in service systems. This is understandable. There is much confusion about the definition of service and service innovation. In anticipation of this confusion, one of the first components of this text is a discussion of the definitions of service and service systems.

Perhaps you are well-versed in service science, Service Dominant Logic (SDL), Viable Systems Approach (VSA), or other popular theories of service. If so, you should find the descriptions of modeling of service systems in this book valuable extensions of your knowledge that will enable you to apply the strategic principles of service to an operational level. Throughout this text, the connections of these theories to the models of service systems will be explained.

I too seek a better understanding of service systems. After spending 25 years as a researcher in the field of manufacturing and product supply chains, I experienced an epiphany in the directions of more fruitful research through the exposure to recent thought leadership on service. My domain of interest was the well-developed field of optimizing models for inventory planning, scheduling, capacity planning, supply chain design, process design, and product design. My colleagues and I in this field quite smugly viewed service as a special case or by-product of manufacturing. We could not have been more wrong. Once I was exposed to the principles of SDL and the long experience of marketing researchers in the field of service, I realized that all of the elegant mathematical models of *goods-dominant* thinking addressed what could be considered a special case of the complicated and complex conditions of service operations.

At once staggered by the challenges of modeling service systems, I was inspired by the opportunity that they offered. We are at an exciting stage of human history. In academic, government, industrial, and social circles, recent years have broadened and deepened the realization that service has gone beyond a ubiquitous presence in our lives to become the basis for all exchange and the processes of living healthy and rewarding lives. In the coming decades, innovative people all over the world will advance the science and engineering of service systems to the level of sophistication and utility that the preceding 100 years of development in the fields of industrial engineering and operations research have brought to manufacturing and product supply chains. My interactions with many brilliant and insightful researchers in the field of service science has revealed to me the rapidly expanding compendium of perspectives, interpretive schema, modeling tools, and applications of service. The novitiate will find this material to be overwhelmingly diverse and with a bewildering variety of applications. In order to instigate and encourage practical and sound ventures into innovation, operation, and evaluation of service, this book serves as a primer on the basics of scientific modeling of service for those who are joined in the advancement of this discipline and its practical application.

In this book, I have synthesized a comprehensive and precise understanding of the manifold definitions, structures, and models posited by many authors. My own background in operations management and operations research afforded me the opportunity to build a rough framework for this material. With the intent of placing this framework within the community of service researchers, designers, and practitioners, I wrote this book. In this way, the book is an interpretation of other work and a synthesis of these works into a cohesive representation of service systems. The purpose of this book is twofold:

- To reveal the existence and essentiality of service *systems* in every service.
- To empower the reader with some basic methodologies for describing service systems in forms that are rigorous enough to support *innovation*, *operation*, and *evaluation* of service.

I propose your exploration of service system modeling through reading this book in whole or in part.

What Is the Importance of Service Systems?

Service systems are the mechanisms that make modern life possible and modern economies viable. Jim Spohrer, one of the foremost thought leaders in service, described to me an illuminating exercise to drive home this point. On any day, recall the list of service systems that were necessary for your normal activities. Everything from electric and water utility services to traffic control, weather forecasts, entertainment services, and of course, Internet resources are provided by amazingly reliable service systems (Maglio et al. 2009). The list is impressive and demonstrates that even the most mundane activities of our lives are made possible through literally dozens of services.

The majority of first and second world economies is based on service. According to the United States Bureau of Labor Statistics more than 85 percent of the U.S. labor force is working in the service sector (Bureau of Economic Analysis 2015). Internationally, the service sector of most economies accounts for more than 50 percent of the GDP and is rising (CIA 2015). These percentages are increasing. Furthermore, service enterprises provide employment at both the lowest wage scales (e.g., hospitality, food service, sanitation) and the highest wage scales (e.g., consulting, education, health care). National economies are now in a global competition for service and, as with the manufacturing economies, winning this competition will be based on two dimensions of performance:

- Improvements in efficiency and effectiveness of service offerings.
- Innovation in new service offerings.

Pursuing both of these initiatives in service will require a technical understanding of how service systems work, and competition will demand ever increasing sophistication in this understanding (Karmarkar 2004).

Throughout the book, we will keep in mind a variety of examples of service systems. We will choose a few examples that are accessible to every reader through common experience. Our examples will also cover the range from basic, low-tech service to knowledge-intensive, high-tech service. We emphasize that service can be both a low-wage and a high-wage enterprise and that all service will be challenged to innovate and improve in the global economy.

Knowledge-intensive business service (KIBS) and *knowledge-based intelligent service* (KBIS) are common types of service systems. We will place special emphasis on KIBS and KBIS as these forms of service are becoming ubiquitous at all levels of service sophistication. Certainly, the IT services provided to business and government enterprises form the industry that garners most of the attention of service innovators and will continue to be a major economic driver throughout the world. However, even the "apps" that people use on their cell phones to serve the most mundane daily activities are forms of KBIS, and we must keep these kinds of service in mind as enterprises that are worthy of innovation and improvement.

You and this book are components of a service system. This book is designed as a knowledge resource with which you can create value in yourself through a deeper and broader understanding of service systems. The manner in which you and I (through this book) interact in creating this value is the subject of this book. Hence, a useful exercise for the reader is to apply the principles and methods described herein to build a model of the service system that we have initiated with your reading. If this exercise stretches your imagination and makes you question the claim that you and this book are components of a service system, so much the better. The method of any worthwhile education service is "to calm the disturbed and to disturb the calm."

Science and Innovation

We like to hear stories of tinkerers who, without the benefit of scientific education, became fantastically wealthy by stumbling on a landmark invention. Admittedly, many innovations came about through uninformed and sometimes daring trial and error. The service economy has been built largely in this way because service is starved of scientific models to guide innovation. The intransigent inefficiency of many service industries such as health care, education, and IT consulting bear testimony to the lack of coherent, model-based methods for service provision (Barthold et al. 2004; Garber and Skinner 2008; Krigsman 2009; National Bureau of Economic Research [NBER] 2015; OECD 2014). Although one can introduce new products and services in this way, long-term success rates are bolstered by knowing what you are doing. This book outlines the state of service science in terms of the basic model constructs that have been derived to date. These constructs are sufficiently developed to allow significant progress in model-driven design of service systems.

Consider the history of the automotive industry. More than 150 years ago, scientists achieved a basic understanding of the chemistry of combustion and the Carnot model of thermodynamic cycles. Mechanical engineers applied this science to invent the internal combustion engine and other automobile components. Process engineers applied various sciences of human factors and mechanics to design manufacturing processes for building automobiles. Business managers across the functions of finance, marketing, and operations applied the sciences of economics, decision analysis, psychology, and physics to design and build supply chains, distribution channels, retail operations, and customer support functions for automobiles. Technicians applied the product and process engineering to learn how to build, maintain, and repair automobiles. Workers in all functional areas applied the systems that managers had designed and built in order to produce, deliver, and sell automobiles. Ultimately, drivers operated the vehicles that were produced by the industry. Hence, the automobile was commercialized by shrewd business managers such as Henry Ford and put to valuable use by millions of car owners. Once commercially successful automobile companies were launched, the industry continued the application of science through engineering and management to measure and evaluate performance of these companies and to continuously improve product, process, and business design.

Similarly, the science of DNA has led to an ever-expanding catalog of sophisticated pharmaceuticals for the treatment of disease, the science of electricity and magnetism led to the creation of logic circuits and the IT revolution, and the science of economics led to the various functional disciplines of business management.

The story is the same in all industries—science enables engineering, which enables management, which enables production and use. An industry can be viewed as a chain of expertise from scientists to engineers to managers to technicians to operators and end users. The technologically advanced society in which we live has been made possible by the engineering of solutions to practical problems by applying scientific models.

Why Do We Need a Science of Service?

The mechanical engineer in the automotive industry designs the product and the output of this effort is represented by the product's blueprints. What is the analog of the mechanical engineer in a service industry such as consulting, education, health care, and tourism? What is a service engineer, and what does a service blueprint look like? What expertise does a service engineer need? How is designing a service different from designing a product? What science does a service innovator apply? This last question brings us to the subject of this book. The science of service is nascent but has achieved enough progress to offer the designer a suite of useful principles, techniques, and models (Demirkan, Spohrer, and Krishna 2011; IBM Research 2004; Maglio, Kieliszewski, and Spohrer 2010a; Spohrer and Maglio 2008).

Managing a service means planning and controlling service. Continuing the analogy of managers as "enterprise engineers," the management of service requires planning and controlling the resource integrations that execute the service. As much as the manager of an engine assembly plant must understand the manufacturing process and the product specifications, the manager of a service system must understand the value proposition of the service and the processes by which the service is created.

Our understanding of service is inadequate for the design and management of service (Chesbrough and Spohrer 2006; Ng et al. 2012a; Ostrom et al. 2010). Currently, the science of service and the application of that science have not promulgated the body of knowledge that enables reliability, efficiency, and effectiveness of service that is on par with these performance measures for manufacturing. Manufacturers are accustomed to the use of computer-aided engineering systems for designing products to precise specifications, process design that achieves near-perfect quality and world-class efficiency in competitive markets, decision support systems for operations planning and control that achieves cost performance within a few percent of optimum, and work forces that continuously improve processes based on thorough understanding of process parameters and technologies, as well as customer requirements. Service industries cannot claim such performance.

How did the design and management of manufacturing supply chains achieve the level of sophistication and performance that we see today? It all began more than 100 years ago with the industrial revolution and the ensuing development of the disciplines of Industrial and Systems Engineering and Operations Research. From rudimentary models of manufacturing processes to today's computerized decision support systems based on mathematical models, these disciplines have discovered and applied a science of industrial systems. The development began with science, which enabled engineering, which enabled management (marketing, operations, finance), which enabled operation of manufacturing systems and product supply chains.

Where is service in this evolutionary path? For many decades, it was thought that producing service is simply a variation on the theme of producing products. Only within the past decade has the realization that service is radically different from manufacturing taken root within academic research communities and within corporate strategy rooms (Maglio, Nusser, and Bishop 2010b). Along with this realization has come the discovery that service operations and management are highly inefficient and ineffective compared to performance of manufacturing systems. Furthermore, service engineering finds itself applying only basic design principles because the science of service is in an early stage of development and its principles are not widely understood (Ostrom et al. 2010; Rust 2004; Spohrer et al. 2007). This book encourages this development by presenting the most practical constructs derived to date from service science and offers them for application and refinement by service designers, managers, and operators.

Creativity, the ubiquitous demand on employees in all modern enterprises, is often put forth as an excuse for rejecting any attempts at imposing structure, and the service innovator who suggests applying the principles described herein is likely to encounter this excuse. The logical response to such resistance is both obvious and disarming. Principles, methods, and tools that provide a framework for service innovation have the power to enable and *leverage* creativity by mitigating miscommunication, problem misspecification, and the inability to learn from experience. As designers of service systems, we are enabled by modeling tools to give us a framework for expressing our creativity. Going forward with the reading of this book, let's look for this leverage and, wherever definitions, principles, or methods appear to be invalid for the reader's sphere of reference, let's try to identify the specific discrepancies and determine the bounds on the usefulness of the modeling.

Modeling and Science

Modeling brings a scientific approach to service. Our interest in modeling is a direct effect of our interest in a scientific approach to service innovation, operation, and evaluation. Modeling is the core business of science because models express relations that are the foundations of all scientific knowledge. Every scientific discipline from biology to physics and from sociology to anthropology grows by creating ever-more accurate and useful models of their domains of study. Scientific models can be mathematical or conceptual, rudimentary or amazingly complicated, but they all serve the same purpose of providing explanations for natural phenomena.

Models are never perfect. Nevertheless, models are useful for finding solutions to problems even if these solutions cannot be considered optimal. As science derives better models, better solutions are enabled, but the problems of any time, many of them "wicked" problems, demand the best solutions at our disposal, and cannot wait for a complete science to reveal the ideal answers. Wicked problems are easy to find in the world of service—designing smart cities, achieving cybersecurity, providing cross-cultural social services, ensuring affordable health care, building cognitive assistants for high-level decision support, establishing sustainable energy supply and demand, supporting aging populations, … the list goes on. The science of service is far from being well developed. Key foundations of this science, such as the definition of value and the mechanics of its cocreation by multiple parties in a service system are yet to find a broadly supported doctrine. Service scientists from many different disciplinary backgrounds continue to research and debate these issues. However, no science is ever complete, and there comes a time in the history of every science when enough theory is established to enable some practical engineering. This book was motivated by my belief that we are now at this point in the study of service and that we can enable and encourage explorations in the practical design of many service systems (Gronroos 1994).

To wit, this book reviews service science research and posits concrete definitions and postulates about service systems. Some of these assertions are too narrow to satisfy the entire service science community who will certainly find exceptions to the structure that is posited herein. Some of these assertions may run counter to widely held beliefs or understandings within the service science community. With apologies to my esteemed colleagues, I have set down in these pages a framework for modeling service systems that requires precision where there has been ambiguity and a single direction where there continues to be alternative viewpoints. The justification for this hauteur is the opportunity to provide a basic toolkit to service innovators with which substantial progress can be made in *innovation, operation*, and *evaluation of service.* Thanks to the accomplishments of service scientists, I believe that enough is known about service to build an engineering discipline around this subject. To be sure, future scientific research will alter and even replace some of the principles laid down in this book. We can look forward to this enlightenment.

In those instances of the need to select a well-defined point of view in the midst of an open scientific debate, I have entered sections in the chapters of this book titled “Re-Thinking.” These sections compare and contrast prevailing concepts in service science and posit the rationale for the position that I take in this book. Depending on the reader’s background and exposure to the very broad subject of service, some of these rethinkings may not involve any rethinking at all as the concepts under discussion have not found a secure home in the reader’s perspective. In other cases, the rethinking may challenge a reader’s cherished principles of service. Perhaps, my representation of service and service systems will illuminate new paths of reasoning. If not, I hope that my explanations at least serve to justify a representation of service and service systems that supports useful modeling and is robust enough to admit the variations in definitions or principles that other researchers feel compelled to adopt.

Structure of the Book

We are about to create an understanding of a complicated subject. We will build this understanding incrementally. The remainder of this book is organized as follows. Chapter 2 cultivates an understanding of the essential defining characteristics of service. The reader who is familiar with SDL and has an appreciation for service as a cocreative activity may find a brief skimming of this chapter adequate. Chapter 3 gets into the meat of modeling by rigorously defining the core elements of service systems. Chapter 4 defines the structures that incorporate these elements into service systems. With these modeling constructs defined, we can turn our attention in Chapter 5 to graphical modeling methods that have proven to be very useful in service modeling. By this time, the reader will understand the key role of agent decision making in the trajectory of a service, which leads to Chapter 6 on decision making. As decision making executes each stage of a service process, we cannot model a service system without modeling decisions. In Chapter 7, we lay the groundwork for decision modeling. We will illustrate the concepts and arguments presented herein with several examples that encompass a range of service implementations from basic to complex.

Summary

- Service is the core function of economies throughout the world.
- We seek to thread science, innovation, operation, and evaluation in a coherent, mutually supporting sequence.
- We apply a science of service in order to create service innovations that are useful, practical, effective, and efficient.
- Modeling is the core function of science and models are the tools of innovators.

CHAPTER 2

Preliminary Concepts of Service

Seeking a Definition of Service

The starting point for a science is the definition of the subject under study. In the case of service science, this definition requires clarification because the explanation of service itself does not have a universal acceptance. Unfortunately, the conventional definitions of service are misleading and inadequate.

We seek a definition of service that makes sense to a modeler of service. A model begins with a definition of a system. A modeler must have a scientific basis for identifying the structure and process of the system under study. These basic elements of a model are derived from a knowledge of the essential purpose of the system and the scientific understanding of the system. Unfortunately, there persists much confusion in the research community about this purpose and science with profound effects on model formulations. Hence, we posit in this chapter a definition of service that has formed the foundation of modern service science and for which the modeling methods in this book are designed.

What Service Is Not

Most textbooks characterize a service as an operation that generates outputs that are intangible, heterogeneous, instantaneous, and perishable—the *IHIP* definition. Sampson and Froehle (2006) convincingly demonstrated the shortcomings of each one of the characteristics as defining features of service. The advertising campaigns of most products clearly indicate that products have valuable intangible features. The laptop computer on which I type these lines is a ubiquitous example of mass customization that has brought extreme heterogeneity to many of the products in use today. By contrast, some services, such as garbage collection, are not very heterogeneous. If instantaneous production and consumption is an earmark of service, then how can we include in the service domain professional services such as legal service, education, health care, and consulting? If perishability is a distinguishing feature of service, then how is it that the education service that you received years ago in high school or university continues to be consumed by you and continues to deliver value to you long after the car you drove when you were in school has gone to the recycling yard? In short, the IHIP definition of service is the product of a superficial analysis of service that took root many years ago in academic research circles and that has proven very difficult to eradicate. A modern service modeler cannot afford to be hampered by this definition.

Conventional NAICS, SIC codes are also rather useless in defining service. Government SIC codes and other forms of categorizing economic activity are widely used throughout the world for publishing economic reports to measure the performance of sectors of economies. However, these categorization schemes are also the products of conventional thinking and historical practice that bear little correspondence to the distinguishing features of service.

Not only do we have a problem in distinguishing service from manufacturing, but we have a problem in distinguishing manufacturing from service. For example, models of conventional service industries such as transportation or food service are very similar to models of product supply chains and manufacturing operations. The basic elements of a model of a product supply chain are resource capacities, process specifications, and material inputs and outputs. For example, from a modeler's perspective, a model of a railroad transportation service looks very similar to a model of manufacturing process. In both cases, material, labor, and machine resources are applied to well-defined processes to transform material from one state to another in a serial, arborescent, or convergent network of such processes. Similarly, a restaurant producing meals from a standardized menu is little different from a manufacturing operation.

We seek a definition of service that distinguishes the features of a service from the conventional view of a product and of a service system from a manufacturing system.

Another popular view of service considers service operations to be special cases of manufacturing operations or by-products of manufacturing. The special-case viewpoint is ignorant of the fact that service is more complicated and more complex than manufacturing. In fact, models of manufacturing systems should be viewed as special cases of models of service systems. Consider a service system such as health care at a clinic. Figure 2.1 shows a simplified graphical model of this system with flows of patients and medical personnel, decision making, personalization of processes, and contingency structures—in short, the variety, variability, and indeterminism of a typical service system. Now let's make four simplifying assumptions.

1. Assume that the processes are allowed to have resource inputs and outputs only in the form of labor, machine effort, and material. Information resources (other than work orders and process instructions) cannot be used to mediate a process or to represent the value-added outcomes of a process.
2. Assume that each patient falls into a category for which the experience of the patient will consist of a standardized sequence of standardized processes for all patients in that category.
3. Assume that the time, effort, input resources, and outcomes of each process may be stochastic and time varying but not vague or ambiguous.
4. Assume that the paths of patients through the clinic from one work station to another are predefined according to categories of patient treatments.

With these simplifying assumptions, the model of the clinic becomes that of a multicommodity, multiechelon, stochastic, nonstationary supply chain—one of the most challenging supply chain models to describe and optimize, but only a special case of a service system.

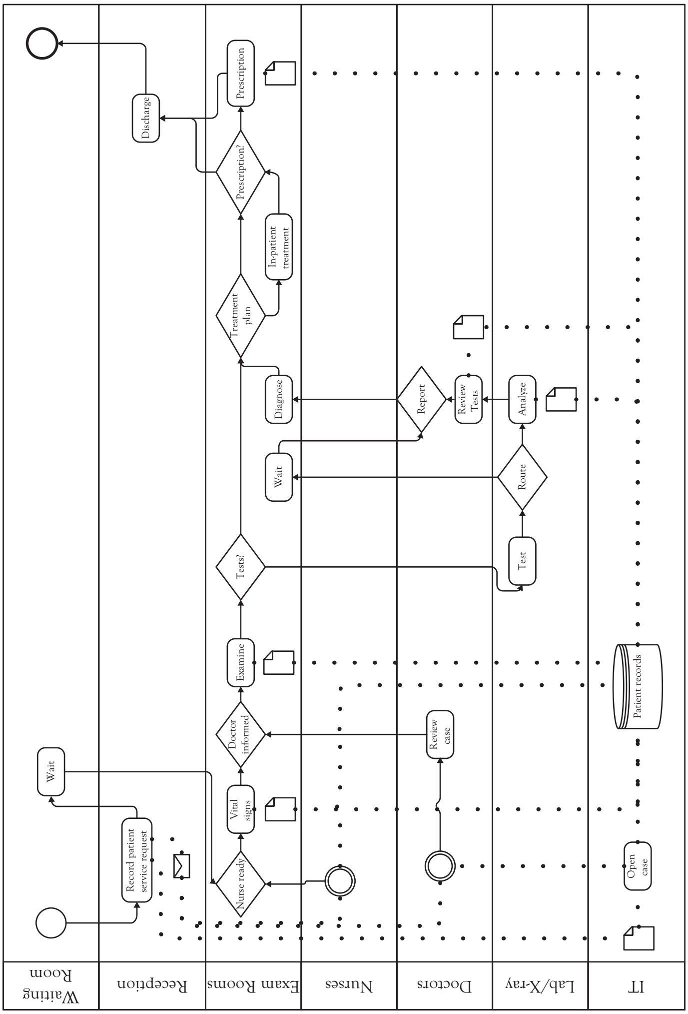


Figure 2.1 Medical clinic

The SDL Theory Initiated Useful Definitions

In the 21st century, a new paradigm for defining service took root in the form of a theory known as Service Dominant Logic (SDL) (Vargo and Lusch 2004). Within the marketing discipline, the IHIP definition of service was recognized as deficient by many researchers and practitioners. These concerns culminated in the publication by Vargo and Lusch in 2004, which set down a coherent set of postulates for service which form the original 10 Foundational Premises of SDL, listed in Table 2.1. Since the first publication of these premises, Vargo and Lusch and an ever-expanding legion of researchers have refined, extended, and applied the SDL theory. For the uninitiated reader, some thoughtful reflection on these postulates is due as they have instigated a revolution in thinking about service in the past decade.

SDL reveals the key deficiency in the IHIP definition of service and other conventional perspectives on service by asserting that cocreation of value is endemic to all service. FP1 to FP6 of Table 2.1 compel any definition of service to recognize that cocreation of value is the fundamental distinguishing feature of service and, hence, must be the essential specification in any definition of service. Furthermore, SDL reveals that value-in-exchange, the monetary transfer that takes place at the purchase of a product or service and is recognized in economic data as value, is not the focus of service or service innovation. Instead, value-in-use and

Table 2.1 Foundational premises of SDL

FP1	Service is the fundamental basis of exchange.
FP2	Indirect exchange masks the fundamental basis of exchange.
FP3	Goods are a distribution mechanism for service provision.
FP4	Operant resources are the fundamental source of competitive advantage.
FP5	All economies are service economies.
FP6	The customer is always a cocreator of value.
FP7	The enterprise cannot deliver value, but only offer value propositions.
FP8	A service-centered view is inherently customer oriented and relational.
FP9	All social and economic actors are resource integrators.
FP10	Value is always uniquely and phenomenologically determined by the beneficiary.

value-in-context are the relevant measures of value (Chandler and Vargo 2011; Vargo and Akaka 2009). The definition of service that we adopt for our investigation of modeling service systems is a slight variation on Vargo and Lusch's definition:

Definition: *Service is an activity initiated and mediated by two or more actors through which value is cocreated for these actors.*

We slightly rephrased the SDL definition of service in order to emphasize the cocreation of value for all service participants as opposed to conveying the notion of service providers and service recipients. Service cannot be viable unless all parties involved in the service extract some value from it. Hence, we insist on the pursuit of mutual (not necessarily equal in magnitude) value gain as a requirement for service. Other definitions of service that identify cocreation of value as an earmark of service, still constrain the definition by requiring a provider–recipient relationship in every service through which the provider enables value creation of the recipient. We broaden this view by recognizing that all actors engaged in a service must derive some value from it, and the roles of provider and recipient can be indistinct.

Furthermore, we do not constrain the definition of service to cases of only two participants. As we will see, modern service systems often employ numerous actors in complicated networks of interactions.

Service and Context

The ramifications of this definition of service are far-reaching and intriguing. Pressing the definition to its ultimate implications, we are forced to conclude the every economic endeavor is a service. The recognition of this fact is spreading across many industries. Service-oriented architecture is infiltrating the world of systems development (Demirkan 2015; Demirkan and Delen 2013). Even products can be viewed as encapsulations of service potential, and this realization has stimulated the servitization (also known as servicization), which is sweeping the manufacturing world (Baines and Lightfoot 2013; Kastalli and Looy 2013; Lay 2014; Vandermerwe and Rada 1988). Returning to our examples of transportation and restaurants, we can see how these operations can be viewed as value cocreation operations if we look beyond the obvious material flows. For example, in rail transportation, the transporter and the client cocreate value by sharing information regarding rail capacities and delivery constraints in order to negotiate a mutually beneficial shipping schedule. Restaurants that combine Internet-based personalization of meals or meal service go beyond the manufacturing-plant view of food service.

Of particular importance to the service modeler is the essential role of context in value cocreation. A profound implication of FP10 in Table 2.1 is that cocreation of value, being subjective and personalized, is characterized by hypervariety over contexts (Ng 2014). By definition, every service engagement is devoted to the creation of value for a unique human actor. Consequently, the dimensions of value, the scaling of value, and the determinants of value are subjective and context dependent. For example, each student in a course has his or her unique educational desires and needs which determine the nature and extent of the effort expended by the student and the outcomes of the course in terms of knowledge and skill creation of the student. This example illustrates the extreme challenge of every service system to evaluate and adapt to the context of every service engagement. Hypervariety across instances and hypervariability over time are endemic to service, and the service modeler must consider context as a fundamental element of any valid representation of a service system.

Service Innovation

Another term that has become both trendy and confusing in recent years is service innovation. What was the motive behind the introduction of service innovation as an activity distinct from service design or service engineering? There are three aspects of innovation that give it meaning in service.

1. *Invention is not innovation*. There are many inventors but few innovators. Inventors design the technical specifications of products and services. Inventors often allow themselves to create their designs without adequate consideration of the needs, wants, and delights of all stakeholders in the value chain. By contrast, innovators know

how to commercialize products by designing processes—the process through which an actor derives value-in-use from a product, the process that a manufacturer executes in order to produce a product cost-effectively and with high quality, and the processes of marketing, finance, human resources, and information technology (IT) which must be marshaled and coordinated to bring the product to market successfully. Famous innovators such as Thomas Edison, Henry Ford, and Steve Jobs made history through commercialization, implementation, and deployment of inventions. The skills and motives for innovation versus invention and process improvement versus technology installation form a strong barrier that separates inventors from innovators.

2. *Innovation is cocreative*. From concurrent engineering to agile software development, innovation has been understood to be a cocreative activity. More than 40 years ago, an idea known as concurrent engineering took hold in manufacturing industries and revolutionized the process of designing products. In retrospect, it seems obvious that the design of a product should be made in consultation with the end users of the product and the people who have to manufacture and service the product. However, this team-based approach to product design with the voice of the customer (VOC) guiding the design process was a wrenching change for many engineering staffs who were accustomed to a sequential design procedure through which design engineers initiated the design process and all other stakeholders, including end users, had to adapt their use cases to the product. Nowhere were the shortcomings of this approach more evident than in software development, which amassed an embarrassing and costly history of project failure (Krigsman 2009; Wailgum 2009).

It is not surprising that the IT industry has rediscovered concurrent engineering in the form of agile project management. Better-quality services with lower life-cycle costs and shorter development times are the outcomes of successful implementation of the concurrent or agile approach to design. Fundamentally, the benefits of the concurrent or agile design approach derive from a design project that is guided by cocreation of value-in-use of a product or software, which, in essence, describes a service.

3. *Service innovation is customized.* Conventional products may not be customized, but inevitably their value-in-use is contextualized by the customer to value-in-context. Recognizing this fact, every product is nothing more than an encapsulation of service potential, with the value of the product derived from the use in context of the product by the customer. In other words, value requires a verb (Ng 2014). Value is generated only through an action or a process undertaken by the user of the product in that user's unique context. Taking this argument to its ultimate conclusion, we see that all design should be managed as innovations instead of inventions. As service is, by definition, a cocreative process, every service is both designed and executed with the participation of all stakeholders in the service. Hence, service cannot be designed or invented by an engineer acting in isolation. The expertise of the service engineer must be married with the resources and desires of all service actors in order to cocreate value. This requirement makes service context dependent, and its value to each service actor is uniquely determined by that actor.

Service innovation as service has become a tenet of service science, and researchers around the world have crafted various frameworks for such innovation. Hastings and Saperstein (2014), for example, have advanced the specification of service innovation through a Seven-Point Service Thinking Framework.

- Cocreation of Value
- Service = Experience (Empathy)
- Service Systems
- Modular Business Architecture
- Global–Mobile–Social Scalable Platforms
- Run–Transform–Innovate
- Multisided Metrics

From the interconnection among these components of service, each of which is complex and complicated itself, it is clear that service innovators need to be service systems modelers.

Summary

- The distinguishing characteristic of service is cocreation of value.
- Every value-creating process is a service.
- Service is characterized by hypervariety and hypervariability.
- There are many inventors but few innovators.
- Cocreation of value requires service innovation and service innovation itself is cocreated.

CHAPTER 3

Modeling Cocreative Systems

Service science is the modeling of cocreative systems (Maglio et al. 2009). In any field of study such as biology, physics, chemistry, psychology, economics, marketing, and management, there is a science from which the field's engineering, technology, and operations are derived. Every science precisely defines the scope of a field of study and the relations among the elements that fall within this scope. Physics defines the particles that make up all matter, the fields that bind this matter together, and the laws of motion that determine the trajectory of physical systems. Biology defines the cellular, organic, and systemic components of all living organisms, the processes by which they interact, and the laws by which biological systems live. In short, a science establishes valid constructs for engineers to build models of systems within the field of study. Although service has existed for as long as humans have walked the earth, the science of service is relatively new, as the understanding of service as cocreation of value has only recently been formalized. Service science research has accomplished much in the past decade and has posited some fundamental specifications of systems that enable service. For the service modeler, we present these constructs in the remainder of this book.

What Is a Model?

The word "model" is used in a variety of contexts and, frankly, is overused and often misapplied. People refer to data models, organizational models, business models, process models, decision models, product models, and even fashion models. To set the record straight, whether they be physical models, graphical models, conceptual models, or mathematical models, every model is an abstraction of some real system.

For our purposes, a model is a valid and useful abstraction of reality. Whenever sufficient data exists, the accuracy of a model can be established through a process of validation that compares predictions from the model with actual system performance. In other circumstances, the validity of a model must be based on the logic by which it was derived and the reasonableness of the assumptions that defined its scope.

Model construction begins with abstraction and encapsulation of the structure and processes of a real system in terms of standard model components. Researchers in many different modeling domains have developed unique catalogs of classes of objects that can be configured to represent the systems that need to be modeled. Biologists have DNA models, chemists have bonding models, physicists have particle and field models, and operations researchers have decision models. Having a set of standard forms from which to build models within a domain of interest is highly beneficial for several reasons. First, the standard forms provide the modeler with a proven methodology for model building. Second, the standard form provides a lingua franca for all modelers to communicate their designs with one another. Third, and most important, the standard forms represent a valid specification of the basic elements and relations that describe the domain of study. We are interested in building models of service systems. In this chapter, we introduce the definitions of the fundamental abstract components of models of service systems.

Examples of models abound in our everyday decision making. When we face the decision of where to eat dinner, we construct a model. The decision is based on a cause–effect relationship between the choice of restaurant and certain key performance indicators (KPI) such as the cost of the dinner, the enjoyment of the cuisine, the comfort of the restaurant's ambience, the time and difficulty of traveling to the restaurant, and so on. In our minds, we construct an abstract representation of this relationship which then defines a feasible set of options and also allows us to select a satisficing options if not the optimal option. In like manner, we construct models for the selection of a wardrobe, the route to take for a road trip, the interior design of a house, or the time to schedule a visit to the dentist. In our working lives, we encounter models when we choose an investment plan, set a work schedule, promote a product, design a service, improve a process, or make any decision that affects the future performance of the enterprise. Formal models for decision making have been under development for more than 50 years, but in the past decade, the development has turned toward modeling service (Badinelli 2010, 2012; Barile 2009; Sampson 2015a).

There are many kinds of abstraction. It is likely that you have seen models in the form of flow charts and diagrams. These graphical depictions of objects and relations are ubiquitous in all modeling domains, and in Chapter 5, we will review a catalog of ones that are particularly suited to service systems. However, an abstraction of a real system can also be made with precise definitions of variables and mathematical formulas that express the relations among them. Similarly, spreadsheets and computer programs that convert input data to useful output data are models of variables and relations. Even verbal descriptions of processes, personnel relationships, job responsibilities, and company policies can be considered abstract representations of real systems. The point is that models come in many forms and the modeler should invoke the form that best suits the purpose of the model in terms of completeness, validity, clarity, and model-building efficiency.

Now that we have a rough idea of what a model is, what about the process of modeling? How are models built? It turns out that model building is a rare skill. This perplexing fact is the motive for this book. One often encounters practicing managers and even engineers who struggle to identify opportunities and benefits of models and to proceed rationally through model development. Nevertheless, several decades of teaching modeling techniques to a wide variety of learners, from senior executives to undergraduate students, has demonstrated to me that anyone can learn basic modeling skills.

Procedurally, model development proceeds through three stages: model specification, model validation, and model estimation. Validity is the measure of the correspondence of the model to reality. Every model differs from the real system that it portrays. One should never ask, "Is this model correct?" We use models to make forecasts, evaluate options before making a decision, and prescribe a course of action. Therefore, the appropriate questions is, "Is this model useful for its purpose?" Note that we do not claim that a model is a *perfect* representation of reality, we care only that the model is a *useful* representation of reality.

Every model has a purpose. I have to make this point rather strenuously because model building is often a confused, disjointed team effort with no clear destination. In today's world of big-data analytics, the danger of misguided modeling appears to multiply as rapidly as the volume of data. Model building is a project. Before, during, and after a model-building endeavor, the purpose of the model needs to be clear to the project leader and the project team.

Models of service systems are useful if they help us innovate, manage, and control services. These purposes take the modeler through all levels of detail in abstraction of a service system, from high-level, strategic outlines of the service to minute-by-minute or even second-by-second descriptions of service processes. Then what is the scope of service system models? People, agents, data, information, authorizations, communications, transformative actions, machines, and venues are all elements that can be abstracted into a model of a service system. Therefore, more than one modeling system will be needed. We will need different model forms for different purposes, but for any service system, there must be consistency across all the representations of the system. Consistency is as important as validity.

Let's consider some illustrative examples. Health-care service comes in many forms and is effective through many different kinds of cocreative opportunities. Strategic models for health care include the actuarial formulas for insurance companies to use for setting premiums, capacity--planning formulas for sizing hospitals and clinics, and investment planning formulas for committing research and development (R&D) budgets. These models are best represented as mathematical or computer models. Tactical models for health care include layout diagrams for improving the efficiency of patient flow in clinics, vehicle-routes for providing emergency response in minimum time, and logistics plans for vaccine delivery (Finkelstein et al. 2015). These models are represented with diagrams of floor plans and geographical areas, backed up by databases of parameter measurements and mathematical models of KPIs as functions of alternative system structures. At the operational level, health-care models include flow charts for performing medical procedures, tables of shift assignments for hospital personnel, and graphical computer simulation models of patient flow in clinics. These models are also backed up by databases of parameter measurements and mathematical models of KPIs as functions of sequencing and assignment choices.

There are two features of these models that are worth noting. First, every model incorporates patients and providers—health care is always cocreated by these two parties. Second, mathematical modeling is inescapable if the purpose of the model is to make hard decisions. However, graphical displays enjoy broad popularity because they go a long way to demystifying the quantitative representation of a service system and are used both to expose the dimensions of a mathematical abstraction of the service system and to illustrate the quantitative measures of performance of the system under different alternative decisions.

Rethinking: Emergence

Throughout all branches of science, the reductionist approach to modeling has been the norm. However, when natural phenomena are the outcome of large and complicated systems, they are labeled "emergent" (Ng, Maull, and Smith 2011). The term emergent is well known in systems science and, as service systems are usually large and complicated, this term is often used to describe the performance of service systems. So far, so good. The problem arises when some service modelers misrepresent the phenomenon of emergence as some kind of mystical effect that defies any attempt at a scientific explanation, fueling their argument against a reductionist modeling approach.

There is no need for a philosophical rift between "systems thinking" and reductionist modeling, as the history of science has amply demonstrated the beneficial and mutually supportive interplay between the two. For example, as scientists began their study of thermodynamics, temperature, pressure, and entropy were observed and measured emergent properties of fluids. However, the quest for explaining the physical world led to the discovery of the atom and molecules which quickly enabled the derivation of the kinetic theory of gases and an explanation of these formerly emergent phenomenon in terms of microscopic processes. Similarly, the periodic table of elements was developed from observations of the macroscopic chemical properties of elements, but then the structure of the atom was modeled, the theory of chemical bonding was launched, and the periodic table was illuminated with an elegant, microscopic model of electron orbitals. Body temperature was known by physicians for thousands of years as an emergent phenomenon of complex (poorly understood) bodily functions, but once the cellular basis for organ performance and the interactive effects of organs on whole body function was better understood, an explanation for body temperature could be made. Radioactivity was first observed as a mysterious emergent phenomenon of some elements, but when a model of the nucleus of an atom as a constellation of subatomic particles was validated, radioactivity was understood as a natural and predictable process of particle decay and conversion. Economist recognized the law of supply and demand as an emergent phenomenon of large populations of buyers and sellers, but when game theory described the process by which two parties engage in business transactions, economists had a detailed explanation for the law. In all of these cases, a macroscopic, systems view of a domain of study yielded an initial specification of a science in terms of unexplained (complex) phenomena. What followed through the natural course of scientific inquiry was an unraveling of the complexity of the phenomenon through reductionist modeling of the system under study. In effect, complexity was replaced by complicatedness by pursuing emergence with reductionism. We can see this process as the mechanism of healthy science. Therefore, systems thinking reveals the emergent phenomenon that reductionists attempt to explain. Both approaches to modeling are not only necessary, but they must be litigated in a mutually supportive manner in order to ensure validity and relevance of the scientific process. We will embrace this marriage of systems thinking and reductionism in the modeling methods that this book prescribes. In the rest of this chapter, we will define the smallest elements of our service systems for the reductionist view. In later chapters, we will construct the "service molecules" from these elements and, from these in turn, models of service systems.

What Is a System?

The next natural question is, what is a system? Talk about an over-used word! In common speech, we toss around the word "system" like a catch-all term, but because we wish to model systems, we need a precise definition. According to the Merriam Webster Dictionary (http://www.merriam-webster.com/dictionary/system), a system is "a set of independent, interacting components forming an integrated whole." OK, this definition informs us that a system has components that may themselves be systems. Hence, systems can have subsystems.

Interaction is another key feature of systems. Maglio and Spohrer (2008) defined a service system as "value co-creation configurations of people, technology, value propositions connecting internal and external service systems and shared information." In other words, a system is not a system unless the components somehow influence each other. Interaction requires more than just a relationship among components. Interaction implies dynamic processes that transform, create, or destroy system components. For example, a family can be considered a system of components with a hierarchical relationship among them (parents and children). However, the parent–child relationship has no meaning except through the acts of parenting.

The broadest perspective on systems defines a system as a collection of interacting components within a larger universe (Bertalanffy 1972). A system, therefore, has a boundary between itself and an "outside world." Furthermore, the system may or may not be able to interact with this outside world leading to the definitions "closed system" and "open system," respectively. As systems science became formalized with the study of thermodynamics more than a century ago, systems have been defined in terms of thermodynamics principles. To wit,

- An isolated system does not interact with anything outside of the system boundary;
- A closed system exchanges energy across its boundary; and
- An open system exchanges both energy and matter across its boundary.

The problem is, how do we interpret energy and matter in a system such as a health-care service system? Do not be surprised to see some very loose analogies to energy and matter in the service science literature. Fortunately, we can almost always view a service system as an open system under anyone's definition of openness because cocreation of value for people inevitably involves many-faceted interactions of those people with their environments.

The big picture on modeling systems is that there are two fundamental aspects of every system and, correspondingly, the task of modeling a system necessarily requires two development stages. The two essential features of a system are structure of the components of the system and the transitions by which the system evolves. Various sources define these two components with different names. These features are referred to as structure—process, system—dynamics, state—evolution, architecture—rules, and other combinations of words that describe a set of components that are related in some ways and a motion that specifies how interactions among the component and between the components and the external environment cause the system state to change. For consistency in this text, we will adopt the names structure and process. For example, a healthcare clinic is a system composed of doctors, nurses, patients, examination rooms, and so on. These system components can be defined in terms of their roles in the system and the requirements of their relationships to one another—the system structure. At any time, the state of this system is defined by the positions of all patients in the clinic and the engagements of all staff and facilities. The arrivals of patients, patient data collection by nurses, and patient examinations by doctors are actions that change the state of the system.

Processes in Service Systems

For service systems, we will define service processes as the mechanisms for system transitions. How many ways can system transitions occur? Keep in mind that the word system applies to biological organisms, subatomic particles, and the universe of galaxies, as well as the more mundane lives of corporations, organizations, and service systems. Fortunately, within the more narrow domain of service systems, we can view all transitions of the system in terms of processes. The process model will be a generic and basic building block of our models of service systems.

However cursory your study of service systems has been, you have no doubt discovered that the dynamics of a service system can take many forms such as the routine processing of customer transactions in a bank to the customized intellectual development in a student–professor service engagement. The staggering variety of contexts, actions, and responses that drive a service toward successful completion or failure naturally inspires the service modeler to lose hope of applying some generic language to describe most, if not all service systems. Yet, this is exactly what service science proposes. One of the most essential of these generic modeling constructs is the service process. We will see how a reductionist approach that configures service systems in terms of networks of processes is capable of modeling emergent system performance.

For those who are familiar with object-oriented programming (OOP), a system can be thought of as classes of objects and procedures that modify, create, or destroy those objects. Hence, OOP is a useful framework for understanding the definitions of service system structural elements and service system processes that the ensuing sections and chapters will describe. Of course, OOP is a conceptual framework for system design that is general enough to describe almost all service systems of practical significance. However, service modeling calls for model constructs that are

- Specific enough to the domain of service to enable parsimonious model representations.
- Intuitive enough to enable service providers and clients of all backgrounds to participate in service innovation.
- Basic enough to enable efficient model building with reusable components.
- Modular enough to enable unlimited adaptation for system evolution and robust error detection.

That's a tall order, and although this book cannot provide the be-all and end-all of modeling frameworks, the following sections and chapters will posit many object-based paradigms that go a long way to providing these utilities. The OOP-knowledgeable reader is encouraged to envision how the class definitions for service system structures and the specifications for service system processes could be developed into working simulation models.

Continuing the admonition for rigor in our development of model structures, we should be concerned about precision in our terminology. Every good science defines an ontology for the domain of study, which in turn becomes the language of the designers and practitioners of the domain. A good example is the medical profession using their undecipherable (to the patient) jargon for diagnosis and treatments. In this domain of practice, precision saves lives. In other domains, such as service innovation, precision saves time, money, and engenders a lingua franca for cocreators to merge their talents. Service science researchers have energized the development of an ontology for service. For the purposes of this book, we provide terminology for the most basic and generic elements of service systems. Various initiatives are underway for constructing the desperately needed ontology for service, but we cannot wait for these efforts to reach a fruition that enjoys consensus of support. In this book, we posit the more innocuous definitions of service system elements with the hope that these offerings will not significantly upset positions that have already been taken and will perhaps illuminate some directions for the development of a universal ontology of service.

Service System Structure

We now derive a compendium of generic structural elements for service system models. These elements (classes of objects) are drawn from the perspectives of many service scientists and practitioners who have constructed models of service systems (Mele and Polese 2011). The definitions of the elements given here may differ somewhat from those found in original literature. The reason for this is the desire for definitions that encapsulate all of the generic features of each element and maintain modularity of the elements. Over the decade or so of significant service science research, the properties, boundaries, and functions of these elements have come into sharper focus. A final version of this compendium is still years away, but this book is motivated by the realization that we now know enough about the nature of service systems to advance some definitive and useful principles of service system structures and functions.

Let's be clear about the word "element." We need to define the basic building blocks of service systems. Just as the elements of the periodic table can be combined in infinitely many ways to construct compounds, and compounds can be configured into the structures that we see around us, service systems can be modeled as a network of nested systems. Put simply, service systems can be modeled as systems of systems (Alter 2011). Let's begin at the atomic level. We begin the element list with the participants in the service process.

Actor

Actors are human. Actors are the people who engage in service processes in order to extract value from the service system. Our definition of actor is inclusive. An actor can be a person, family, team, social group, department, corporation, government agency, or any institution of persons with a defined governance structure and categorical values which, in a given context, motivate and guide the behavior of the actor in a service engagement. A service system model must identify any person or organization who is engaged in a service system as an actor. Note that the actors of a service include those who we typically think of as service recipients and those who we think of as service providers (more about this distinction between recipient and provider later). As the distinguishing feature of an actor is humanity, actor elements encapsulate the essential human properties.

Actors define and measure value. We have established that value is the ultimate outcome of a service, but let's delve into this concept of value. By its definition, value is a property of a human being. There is no such thing as value except within the psyche of a human being. Furthermore, value is unique to each person. Actors have the ability to appreciate value according to uniquely personal standards.

Actors defy standardization. Actors are endogenous (Ng 2014), meaning that each actor is uniquely motivated, capable, resourceful, understanding, appreciative, committed, and featured by whatever dimensions of affect that influence the actor's participation in the service system and the way that the actor perceives value. If you foresee frustration in trying to cocreate value for numerous, inimitable actors with a single service system, then you have grasped the fundamental challenge of service innovation.

A service system begins and ends with people. Let's identify some essential properties of human beings and ascribe them to our definition of actor in order to ensure that the element lives up to its billing.

Will

Why can't an actor be a machine? An actor must have motives that come from within. Intelligent computer agents can be programmed to initiate actions, but these actions are not inner directed. By contrast, humans have Will—the internal, often mysterious but always unique initiative to action.

Our definition of actor as human also carries with it an assumption of the nature of the actor's will. An actor's will is guided by some kind of ethics or morality or both. An actor is self-aware and has a conscience. Unless you are interested in designing service systems for the pathologically antisocial or destructive, we can assume that the actors in a service system are not guided by evil or perverse intent.

A word of caution is needed here. A common code of ethics can engender extremely different behavior and decisions across different actors because there is more than will to determining actions and decisions. The context of a decision, which includes personal history, personality, and emotional state, is also a determinant of an actor's course of action.

Actors have "categorical values." This term are taken from a systems theory known as the Viable Systems Approach. See Golinelli (2010) and Barile (2009). Categorical values are the principles, prejudices, standards, or political views that guide an actor's will. These values may be inherited by the actor from institutions to which the actor belongs. Nevertheless, mindful of the unique contexts of cocreation of value, we must model each actor as in possession of a free will guided by categorical values.

Agency is not acting. In the following, we define the element of the agent. An actor may function as an agent or an actor may engage the services of an agent. Under the precise modeling structure that is prescribed herein, an agent (e.g., a computer application or a human representative of the actor) in the employ of an actor behaves strictly according to the actor's will.

Value

An actor must be capable of experiencing and appreciating value. Under our modeling framework, only an actor can experience and appreciate value. Value is created for people, only for people. Value is the outcome of the actor's utility function interpreting the outcomes of a service.

Perhaps you wonder, what is a utility function? Good question. Classical economic theory postulates the existence of a "function" in the psyche of each person, which determines the actor's assessment of value (von Neumann and Morgenstern 1944). Service science has challenged some of the assumptions that classical economists make about utility (Lessard 2015; Ng, Smith, and Vargo 2012b; Sampson 2015b). In considering the role of utility functions in our service system models, we are thrust into a modeling feature that remains under dispute. As I stated in the introductory chapter, our objective is to prescribe a useful modeling structure for the current and near-future service modeler. Hence, we will put aside the explicit modeling of utility functions and direct our attention to the manifest outcome of utility functions, which is value assessment.

Returning to the subject of value, we are not out of the woods. Although modern service science is founded on the principle that the defining characteristic of service is the cocreation of value, many service scientists and service practitioners do not have a clear idea of what value is. For several different perspectives of value, see the special issue of the journal *Service Science* (Maglio 2015) that is devoted to the question of value. It remains the case that value is poorly understood, even by service scientists.

What do we know about value? Examinations of the concept of value for different service participants quickly reveals that value is multidimensional. The value of a meal in a restaurant is derived from the enjoyment of flavors on the tongue, the sense of satisfaction from a filling meal, the warmth of the restaurant atmosphere, the sharing of the experience with co-diners, the memories of relaxing times that the experience invokes, the ego boost of telling acquaintances the next day about the dinner, the amount of waiting time experienced, and many other aspects of the service. These drivers of value span hedonic, intellectual, carnal, spiritual, and every other fundamental dimension of value. These multiple dimensions are not commensurate and combining them into a single numerical value may not be possible. We conclude that value is a multidimensional property of each actor.

Value is dynamic. The value of the restaurant meal can change upon reflection. Different dimensions of the value may increase or decrease over time. Students who may express approval or disapproval of a teacher at the time of completion of a course may come to understand many years later that the value of the educational experience is much different from their early impression. Returning to the points made in Chapter 1 about the definition of service, this phenomenon serves as proof that we cannot define service as activities with perishable outcomes.

Value is derived from a process. Value creation is a process that converts resources into value. We often think that value can be derived by having something, such as a car, a smart phone, or money. For the service modeler, it is critical to understand that a product is an encapsulation of value potential. Only through use in an actor's context can value from a product or the resources generated by a service process be realized or destroyed. With this realization, Service Dominant Logic (SDL) has illuminated key aspects of value by distinguishing among value-in-exchange, value-in-use, and value-in-context (Vargo and Akaka 2009).

A dramatic example of value- in-context comes from the provision of relief services in the aftermath of the tragic earthquake in Haiti in 2010. A relief agency delivered biscuits to a mob of desperate, homeless, and hungry victims of the quake. Quickly overwhelmed by the crowd, the relief truck dropped off a load of biscuits and drove away. In short order, an educated member of the community advised the victims that the biscuits were outdated and would cause gastrointestinal illness if they were eaten. The victims threw the biscuits in the gutter. As it turned out, the date on the biscuit wrapper that was thought to be the expiration date was the production date. In this case, a resource that had literally life-saving value was reduced to trash because of the context.

Agent

Actors have will but not necessarily a lot of intelligence or knowledge. For this reason, there are agents that work on behalf of actors. Of course, an actor can function as his or her agent, but our definition of agent admits external agency on behalf of an actor or a group of actors. An agent can be human or machine. Whether human or not, the agent acts as a decision analyst or process executor for the actor. For example, a real estate agent acts on behalf of a seller and brings experience and knowledge of the market to the service of selling a house. The agent is tasked with producing an outcome of the sale that is valued by the actor for which the agent works. Alternatively, the house owner could post the sale notice through an on-line marketplace in which case a computerized agent is engaged to play the role of real estate agent. Increasingly, agents in service systems are automated, and with the rapid development of cognitive systems, we can expect that most smart service systems in the near future will be built with intelligent computerized agents. Whether human or machine, agents have several characteristic features.

Agents are resource integrators. Agents engage and regulate service processes. The most basic task of an agent is to invoke resources that are needed for the execution of a service process. These resources can be material, data, information, skill, or capacity. For example, the real estate agent might acquire data of the recent sales of houses that are similar to the client's house in order to estimate a reasonable asking price. A computerized inventory control system may order material in response to a customer order. In each of these examples, the agent has access to resources that are needed to carry out one step in a service system.

Agents are decision makers. This potential function of an agent is often overlooked or misunderstood and yet it is the essential function of an agent. An actor is the person for whom a decision is made and the person who determines the structure of value that is associated with the outcome of a decision, but the framing, modeling, and solution of a decision can be made by an agent. Again, the real estate agent provides a good example. Once the house owner (actor) conveys the relative importance of selling price, time to sell, and the owner's limits for renovating the property, the agent can be empowered to identify and evaluate the options for selling and recommend a final solution. In this way, the agent has modeled the selling decision, estimated all parameters relevant to the decision, and optimized the model of the decision. Certainly, intelligent computerized agents have a long track record in providing this kind of decision support to business decisions through what can be called knowledge-intensive business systems (KIBS), and increasingly, as such systems enter the everyday lives of people, they provide decision support through what can be called knowledge-based intelligent systems (KBIS). In both cases, the agent's function is in modeling a decision on behalf of an actor.

Agents are invoked by actors. An agent works on behalf of actors. When the opportunity for service arises, a value proposition is exposed to an actor, which initiates a decision-making process. The actor wants to know, should I engage this service opportunity? The answer to this question could require more time, study, and knowledge than the actor possesses. Hence, the actor calls upon an agent to make the decision or to support the decision with intelligent fact-finding and analysis.

An actor can be an agent. Agents have heads, actors have hearts. Some actors have both. In the case of an actor who has the knowledge, intelligence, and desire to make the decisions that are necessary to engage other agents in service processes, the agent and the actor can be one in the same.

An agent can be nonhuman. Nonhuman agents have become ubiquitous in our lives in the form of so-called APPS. The smart phone and other devices provide a small army of agents to their users for the purpose of engaging or disengaging service opportunities from voice, video, or text communication; vehicle routing; scheduling appointments; making reservations; purchasing tickets; forecasting weather; playing games; to a host of other services. The smart device has revolutionized the service economy.

Role

An agent plays a role for an actor in a context. An actor engages in service many times every day. With every service, agents carry out essential functions such as data collection, data analysis, forecasting, information display, decision analysis, decision recommendation or optimization, communication, payment, proposing, evaluating, bidding, offering, storing, retrieving, accessing, and so on. Every website is an example of an agent working on behalf of an actor and a single website typically embodies several agents that are capable of playing different roles. All of these roles can be necessary at one time or another within a single service for an actor and they require different capabilities. Hence, a single actor retains and invokes multiple agents for different roles.

This definition of Role leads us to a very important element of service that we call Context, which we will define shortly. A role is established by the invocation of an agent by an actor for a particular context. Within a context, the role of an agent is modeled in terms of relationships between actors and the agent.

Resource

We will do a lot with resources in our models of service systems. Simply put, a resource is something that is used in order to make a service process possible and a resource is something that is produced by a service process.

Resources are consumed, transformed, and created through service. For example, the taxi cab is a capacity resource that is required for a transportation service, the credit card number is a data resource that is required for an online payment service, the knowledge of a physician is an information resource that is required for a diagnosis service, and the information on a government website is required for a tax return service. In all of these examples, the resources are input requirements of a service. Resources are also produced by service. The understanding gained by a student is a knowledge resource that is produced by an education service, the treatment plan that is provided to a patient is an information resource that is produced from a health-care service, the meal in a restaurant is a material resource that is produced by a food service, and the on-time arrival of a passenger is a capacity resource that is produced by a transportation service.

Resources can be tangible or intangible. Recall the discussion in Chapter 2 about tangibility. As the examples given in the previous paragraph demonstrate, resources that are associated with service can be tangible or intangible. Value is derived from resources. Value is also an intangible outcome of a service. Some authors have distinguished coproduction, the output of resources, from cocreation, the output of value, perhaps to attempt to distinguish tangible from intangible outputs. However, this distinction is artificial. In the examples given earlier, there are intangible output resources (knowledge, capability) that are not value per se. The value that is created by a service is derived by the actor who receives the resources that the service provides. Therefore, there is no point in separating tangible from intangible resources as service can combine both types of resources as inputs and outputs.

Each resource can have numerous properties. A resource such as a newspaper can be used as an information resource in the service of educating a reader about current political issues. The same resource can be used as an information resource for an entertainment service for a reader who finds political intrigue humorous. The paper on which the political stories are written can be used as a material resource for the services of swatting flies, cleaning spills, and wrapping fish. The point is that a resource can have multiple properties and that some of these properties can be tangible and some intangible. As the newspaper example shows, some properties of a resource may not have any obvious connection to a service process in which the resource is used or by which the resource is created. The relevant properties of a resource are determined by usage in a context.

Access

Agents are resource integrators. The broad definition of resources that we have adopted and the role of agents as the executors of service implies that agents identify, invoke, and commit resources to a service system. Ng (2014) exposed the role of connectivity in the service systems in use today and those that will be developed in the near future. Ownership of a resource is not always necessary for authorizing commitment. For example, information or applications that are available on a website could be vital resources for a service, but the agent that invokes and applies these resources only needs access to them. Access rights to resources can take many forms and in some cases can be quite complicated. For this reason, we include access rights as one of the elements of a service system model.

An agent's role requires certain access rights to a resource in order to utilize that resource in a service process. For example, a real estate agent may have access to a homebuyer's income information when playing the role of purchasing agent, but when playing the role of seller the agent would not have access to this information resource. Hence, access rights to a resource are a function of agent role. Access rights are modeled as relationships between agent roles and resources. This fact brings us back to the all-important notion of context.

Activity

Irene Ng (2014) made the point that value cocreation is a verb. Service transforms input resources into output resources and value. Cocreation of value and resources is an action, not a state. A service then must be modeled as a transformation. Up to this point, we have used the more colloquial term *process* to describe the overall dynamics of service. Now that we are decomposing service into its atomic or elemental parts, we will adopt the term *activity* to identify precisely the functions in a service that transform resources.

How are transformations modeled? Here we can invoke a long history from many sciences. The notion of an activity enables the modeler to decompose a service into a value chain of transformations. A simple transformation can be defined as an activity that transforms input resources into output resources. Complicated transformations generally succumb to decomposition into sequences or networks of activities. Hence, the activity is the basic building block of a transformation model.

We need a generic and robust definition of an activity as a modeling element. To begin, an activity transforms input resources into output resources and value. In general, the transformation requires some time to perform. Hence, an activity can be defined in terms of the usage requirements of input resources, the yields of output resources, the cycle time of the activity, and the value that is extracted by the recipient of the output resources. Be advised that we are not precluding the realistic possibilities that any of these parameters can exhibit uncertainty in measurement or prediction, variability over time, and variation over different contexts. This last point must be emphasized because a common reaction to the suggestion of modeling human-mediated activities is that precise, scientific analysis of such domains is not possible. However, more than five decades of research and development in the field of operations research has convincingly demonstrated that random variation and variability and time-varying environments has not prevented the development and clinical application of thousands of useful activity-based models.

If activities are the working parts of a service system, what sets the system in motion? The first activity of a chain of activities that make up a service cannot start spontaneously. Furthermore, the continuation of a service from one activity to another must be under some external control. The events that initiate and terminate the activity chains of service are invoked by agent roles in a service system and by external natural forces. We will return to these controls when we define authorizations.

Service activities drive the utilization and production of resources. Figure 3.1 shows a simple illustration of a service activity for a simplified model of the service of reviewing a university application. The necessary input resources and the relevant output resources are defined by the nature of the service activity. As you can see, resources only make sense in connection with activity. Also note that the generic service activity can have multiple resource inputs and multiple resource outputs. As service involves cocreation, the resource inputs can be committed by more than one agent and the resource outputs can be received by more than one

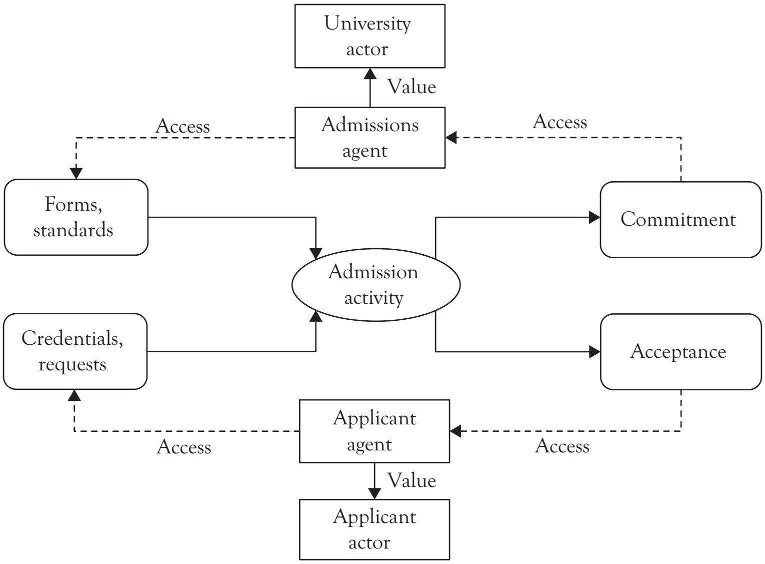


Figure 3.1 Application process

agent. Finally, the value that a service process generates can be realized by more than one actor.

We can make use of the service activity as a modeling element if we define a common set of rules for all activities. The rules are simple:

- Every service activity transforms input resources into output resources.
- Input resources can be provided through access by any number of agents.
- Output resources can be received through access by any number of agents.
- Every service activity requires a cycle time to perform the resource transformation.

Simplicity and precision breed robustness. These rules provide us with a general-purpose building block for the modeling of the complicated and complex service evolution. We can build models of large service systems by networking as many activities as we need. We can capture complicatedness by decomposing a service system into the level of detail that we need.

We want our models to give us a quantitative representation of a service system. So far, we have defined model elements that give us a rich vocabulary for describing service systems. However, the goal of our modeling initiative is the ability to describe and optimize the innovation, management, and control of service systems, and this means building models that quantitatively relate performance measures to the decision options of service agents and the parameters that regulate service evolution. The activity model element is the foundation of our quantitative representation of service systems. Each activity is parameterized by the following data elements:

- The usage rates of input resources
- The yield rates of output resources
- The activity cycle time

How can everything that happens in a service system be reduced to an activity? Let's consider the breadth of representation afforded by the simple structure of the activity element. A transformation of input resources to output resources, with resources defined generally to represent any material or nonmaterial asset, can be used to represent any stage of a service system. In particular, this activity element can be used to represent decisions, as a decision transforms information resources into commitments or authorizations (also information resources). Decisions and other actions that involve information resources can be modeled as activities. Maglio and Spohrer (2008) asserted that service systems perform four basic activities: interact, serve, propose, and realize. We want a modeling framework that can capture all of these activity types and then some. Besides decision making, the activity element can characterize proposing, offering, receiving, information gathering, communicating, questioning, responding, producing, reviewing, messaging, and any other basic step in the evolution of a service. Furthermore, each resource output of one activity can become a resource input of another activity. By connecting activities through their resource inputs and outputs, complicated service systems are modeled. Finally, we can also see that goods-producing activities, which transform material resources into other material resources, are special cases of the service activity element.

Let's consider a simple illustrative example of the element definitions that we have made so far. Consider the case of a trip in a taxi cab as a prototypical example of a service. Figure 3.2 shows how the service activity called a *Ride* transforms the resources of a passenger in a way that adds value to the passenger who is the customer of the service. Suppose a business person needs a cab to take her to an important meeting. The business person hails a cab and the taxi-cab driver, acting as his own agent, dispatches his cab to the rider. The service activity is the ride that deposits that business person to her destination in time for her meeting. The resources that are transformed are the rider's request for a ride (an information resource) to the rider's location in time and space (a state vector) for the meeting.

This example also makes an important point. Figure 3.2 shows a symmetry between the involvement of the cab driver and the involvement of the rider in the service. Who is the service recipient and who is the service provider? Does the rider have a service activity that requires another person to cocreate value by executing the service activity, or does

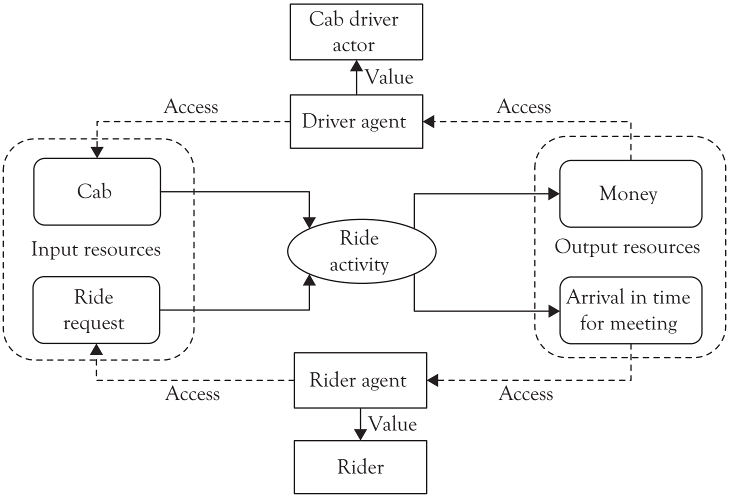


Figure 3.2 Taxi service

the cab driver have a service activity that requires another person to cocreate value? Both participants in the service activity extract value from it. The cab driver receives a monetary resource from which he derives value. The rider receives an arrival in time for a meeting from which she derives value. Did the rider seek a cab driver and engage him in the service activity or did the cab driver seek a rider and engage the rider in the service activity? These questions have no answer. Clearly, there is no point to trying to distinguish client from provider. When we enter a cocreative endeavor, we are both client and provider. To resolve this dilemma, we will use the term "service participant" to identify any actor who engages in a service process.

Activities can have subactivities. An activity that converts resource inputs into resource outputs can be modeled at different levels of detail at the discretion of the modeler. Therefore, we must use the activity element hierarchically so that activities can occur within other activities. Even a simple activity of submitting an application or hailing a taxi as shown in Figures 3.1 and 3.2 can be broken down into numerous steps with associated information resources so that the overall activity is seen to contain a network of component activities. However, even in a large network of activities, each activity has the same fundamental function of converting input resources into output resources.

Irene Ng (2013) defined a contextual archetype as a class of activities that can be invoked for a particular context. Relating the notion of a contextual archetype to the service system structure that we are defining, we can consider an archetype to be a standard or generic form of an activity. Archetypes are very useful to the service system modeler. In many service innovations, subject matter experts or the compendium of common experience illuminates the key features and key resources of a service activity. The archetype then serves as a backbone of a context-based innovation of the service. As a foundation for review, improvement, and customization, an archetype supports rather than constrains innovation, and the service system modeler is well served by a toolkit of archetypes associated with the domain of interest. A simple example is the archetype of serving a meal in a restaurant. Certainly, every restaurant performs this service through some variation on a standard, well-known activity archetype.

Usage and Yield

As every activity transforms input resources into output resources, the activity is marked by the lists of these two sets of resources and the quantities of these resources (in the case of scalable resource types) that are involved. For categorical resource types, the usage and yield parameters are set to one. Usage rates are parameters that represent the quantities of input resources required by an activity. Yields are parameters that represent the quantities of output resources generated by an activity. In most cases, uncertainty about the performance of an activity make these parameters stochastic. Nevertheless, these parameters are essential elements of a model of a service activity.

Authorization

Every process requires some form of formal or informal authorization. An invitation is one form of authorization. A purchase order or a work order is a form of activity authorization. A retail purchase is a activity authorization issued by the customer. To complete the incorporation of the service activity element in a service system model, we need to define authorization as a basic model element.

Authorizations are the instruments of a governance system for service. Authorizations have numerous forms and operate under various context-specific constraints. For example, invitations are offered under different protocols in different cultures. Legal constraints on e-mail spam and automated filtering of e-mail messages effectuate limitations on the tidal wave of invitations to service that most of us receive. Purchase orders, whether for personal or corporate transactions, operate under legal rules for committing to a purchase, financial institution rules for offering credit and transferring funds, and company policies for executing purchase orders. Government service engagements require various forms of citizen consent as well as identity verification. Authorization is an essential but rather complicated and polymorphic class of objects in a service system.

Rethinking: Operand Versus Operant Resources

Students of SDL are familiar with this distinction between operant and operand resources, popularized by Vargo and Lusch (2004). An operant resource acts on an operand resource in the execution of a service process. Operant resources are necessary to instigate and enable a service process. Operant resources are intangible and measurable only in terms of categorical variables. Examples of operant resources are knowledge, skills, and information. By contrast, operand resources are tangible and can be measured with integer or real variables.

Another feature of operant resources is that they can be used to create operand resources as well as other operant resources. For example, in a medical-care service, the physician's knowledge and the patient's knowledge of his or her symptoms and health history are operant resources that, through a service system, cocreate wellness in the patient (operant resource) and days of attendance at work (operand resource).

Unfortunately, there are two aspects of these definitions that lead to model invalidity. First, it is not possible to utilize an operant resource without consuming an associated operand resource. For example, the knowledge that a doctor may impart to a patient can be transferred only through time spent by the doctor in speaking to the patient. The person-minutes of effort that are required of the doctor represent a consumption of the doctor's capacity, which is considered operand resource. Similarly, the commitment of any person's knowledge, skill, or talent as well as the application of software or data from a database cannot be achieved without the expenditure of capacity resources such as CPU time or data transfer volume in a network, however small this expenditure maybe. Therefore, operant resources cannot be utilized without some amount of concomitant utilization of operand resources and these two forms of resources are inextricably connected.

A second issue with the distinction between operand and operant resources is the role of instigator or enabler that is assigned to the categorical-valued resources. Hence, operant resources are naturally assumed to be categorical and operand resources are naturally assumed to be scaled. However, scaled resources such as materials, machine time, messages, and human effort can also be required inputs to a service process, and the presence of these resources can serve to instigate, authorize, and enable a service activity. For example, in some contexts, the presence of a police officer (scaled capacity resource) can authorize and enable a service activity of quelling a domestic dispute by invoking the officer's knowledge of conflict resolution (categorical-valued resource). In other contexts, the officer's knowledge of the prevalence of domestic violence (categorical-valued resource) can motivate her to spend time (scaled capacity resource) visiting a suspected locations of domestic dispute. Then how can we say that operant resources are always categorical and operand resources are always scaled? If we are to construct a modeling framework that is robust enough to model most service systems, we cannot assume that operant resources are necessarily categorical.

We conclude that the more realistic view of resources is that a resource is an object that can have multiple properties, each of which can be either categorical or scaled. Furthermore, there is no such thing as an operand or operant *resource*. Instead, a resource can have *properties* that can perform either an *operant or operand function* in different contexts and that an operant property does not necessarily have to be categorical in nature.

Rethinking: Money Is a Resource

It may seem obvious that money is a resource. However, within the service science community, there seems to be a reluctance by some to view money as a resource. Perhaps the reluctance to consider money a resource stems from the distinction between value-in-exchange and value-in-use, which was brought into high relief by SDL. As money is involved in the exchange stage of a service and use-in-context generally ensues at a later stage, money has come to be associated with a false sense of value, but this impression may not be universally appropriate. SDL has firmly established the principle that value is created during use in context and not during exchange. However, the money that changes hands at the point of exchange is still a resource that can generate value in its own right and can be a resource in support of a value cocreating experience such as a game of poker.

Adherents to the modern definition of service as cocreation of value sometimes do not delineate the resources produced by service from the value created by service. In order to maintain a rigorous interpretation of the definition of value, we impose the condition that resources do not have value. Value is derived from resources, as they are created by a service in a context.

Money is a good example. How does one derive value from money? The obvious answer is that money can be used to purchase other service from which one derives value, in which case money is an input resource to this service. A less obvious answer is that someone can also derive value from money simply by possessing it. Possession is a service. This service can produce resources such as security, prestige, or pride. Deriving value from money or a thing of value, such as a work of art or a fancy car, comes from engaging in a process of ownership such as jingling coins in one's pocket, viewing one's bank balance, storing a work of art in a vault, or parking a car in one's driveway for all the neighbors to see.

Service Evolution

Engagement

Service activities are invoked by engagements, also known as encounters (Bitner, Booms, and Mohr 1994; Qiu 2013, 2014). An engagement element consists of the combined and coordinated access to resources by agents that have been authorized to integrate their resources in a service activity. The outcomes of the engagement are resources that are available to the agents and in turn to the actors who authorized the engagement. These actors assess the value of the resources that are produced by the engagement.

Engagement Decision

Decision activities are particularly important because they drive the service system. An agent is authorized to engage in an activity by a decision. Therefore, decision activities are required for initiating a service and terminating a service.

Context

We now have precise definitions of the components of a structure in a service system that we call a context. Specifying the relationships among the many model elements that we have defined in this chapter, we can construct an aggregated object that we call a context. Be advised that the word "context" is used widely and diversely in service science literature and the definition of context given here is one specific and precise representation of this concept. Like all of the model elements defined herein, we posit this definition from the need for a rigorous specification of the concept of context within the modeling framework that is proposed in this book. We do not disqualify more loose interpretations for other purposes or discussions.

We define a context around the engagement of agents in a service process. Figure 3.3 illustrates a generic context. The context is defined by the actors and agents and a service activity in which they engage. In addition, the input and output resources of the service activity complete the picture. Some features of Figure 3.3 are worth noting:

- The figure is rudimentary because it shows only two actors as participants in the service process, but in general, a context can engage any number of actors and agents.
- Each agent applies its access to relevant input resources to execute the process.

- Each agent utilizes its access to output resources to extract value for the actors and to establish a new state from which the decision to engage in an ensuing context can be made.
- Every engagement in a service activity is preceded by an engagement decision, and these decisions determine the journey of the service. The actors make a joint decision to engage or not to engage in the service process.

We should keep in mind that a service context, as we define it here, does not necessarily constitute a service. As the activity is a single, resource-transforming step, it usually represents only one stage of a multistage experience that comprise an entire service. Even in simple service systems, the engagement of a series of activities includes options for selecting from different available activities within the service system, as one finds, for example, in a retailing website with user-guided selections of item-viewing activities, item-purchasing activities, and bill-paying activities. A visit to a doctor's office for medical treatment would be a service that is produced through a series of coordinated service activities—checking in at the reception desk, waiting, measuring vital signs, consulting the physician, receiving a prescription, and checking out. These six contexts, performed in sequence, form the complete service.

Journey

Engagement decisions commit resources to a service activity. The engagement decision shown in Figure 3.3 involves a decision activity and forms a context of its own (recall the previous discussion about service activities taking the form of a decision). In effect, the figure contains two contexts, one leading to the other. A complete service is formed by a chain of contexts with engagement decisions within the chain mediating the selections of service processes. We will call this chain a service journey.

The frontier of modern service innovation that demands our modeling skills takes us to journeys within large service systems and multiple service systems. Within any service type, the number of possible journeys can become astronomically large due to the road mapping of engagement decisions. Furthermore, the success or failure of a service is determined

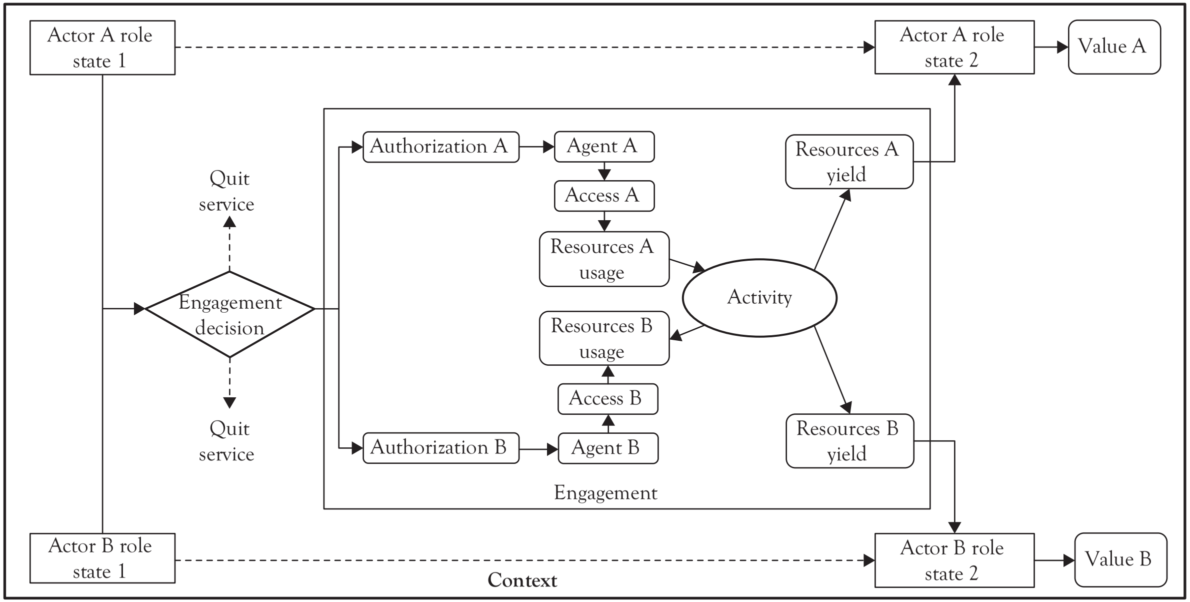


Figure 3.3 A context

by the value accumulated through the journey and the state at which the trajectory ultimately terminates. Therefore, journey modeling becomes the final assembly of our model elements.

Value Proposition

One type of activity is that of offering (requesting) a service by a potential service provider (recipient) and stimulating a response by the recipient (provider). For the sake of discussing the rather tricky topic of value propositions in terms that are more comfortable to the reader, we revert to the conventional service provider or service recipient representation of actors. The key input resource of this offering process is something known as a value proposition, conventionally submitted by the service provider to the service recipient. If the service recipient perceives the value proposition to provide benefits net of the expected costs to engage in the service, the rational recipient will decide to engage the service.

Recent years have seen much research into the most appropriate design of a value proposition. It is generally believed that the design of the value proposition is crucial to the ultimate viability of the service journey. Value propositions provide more than just a catalog of potential outcomes of service. The value proposition should be a mutual commitment of the service participants although most representations to date represent the value proposition as a specification of outcomes of a service that the service provider offers to the service recipient. The value proposition also serves as a guide for the service journey allowing the service provider to manipulate the recipient's journey through those service activities that the provider feels obligated to execute, for example, the execution of a teacher's syllabus, a doctor's treatment plan, an IT support provider's service-level agreement, and so on.

Rethinking: Value Propositions

Service system modelers generally think that the value proposition should be made in the first engagement of an agent with a service system. This prescription is consistent with the view of the value proposition as a type of blueprint or spec sheet for the service that the service provider is about to give to the service recipient. But wait, doesn't this scenario smack of conventional goods-dominant logic? Doesn't this scenario depict the service provider as a manufacturer and the value proposition as an a priori specification of what is to be delivered by the producer to the customer? How can such a perspective of the value proposition find consistency with the service as cocreation of value (not coproduction of resources)? How can this design of a value proposition endorse the reality that the distinction of service provider and service recipient is ambiguous? A strict interpretation of the value proposition certainly raises these questions.

And there is another reason to question the common perception of value propositions. Agile development, a modern approach to software and systems design, is the latest incarnation of an approach to design known as concurrent engineering, which dates back to the 1970s. In the case of software design, outrageous project failure rates over several decades revealed a persistent and thorny defect in the conventional waterfall method of project management, which is the software developer's version of the sequential, design-blueprint-build-sell product development plan that is well ingrained in manufacturing and construction. However, software developers repeatedly are struck by something about their customers that designers in other industries stumbled on rather slowly—the customer does not know what the customer wants. This reality of customer awareness and understanding obviates the utility of any blueprint, service-level agreement, or value proposition. So the agile approach allows the customer (end user of the software) and developer to cocreate prototypes until customer satisfaction is achieved—what a concept!

We come to realize that value is a fuzzy concept. Furthermore, value is dynamic, and an actor's understanding or beliefs about value are likely to change during the execution of service processes. Therefore, not only is the realization of value cocreated over the journey of a service, but the very definition of value is cocreated over the stages of a journey. This is an important distinction. By definition of value, we mean the specification of the dimensions of value and how these dimensions are scaled. Achieving value along these dimensions is a separate issue.

We conclude that a value proposition should be modeled as a work in progress during the lifecycle of a service. Unfortunately, the tradition of goods-dominant product design influences the thinking of service researchers in their modeling of value propositions as "spec sheets" that are given to the service recipient by the service provider. Instead we assert that a more realistic representation of a value proposition is that the value proposition evolves for all actors (let's return to our protocol of not distinguishing providers and recipients) through the service journey. A conventional value proposition, such as a course syllabus, an IT service-level agreement, a vacation tour brochure, a medical treatment plan, an insurance policy, and so on, is an archetype of the value structure that will emerge from the actor-dependent journey of the service.

Summary

- Service science is the modeling of cocreative systems.
- Every model has a purpose.
- Models are not correct, they are useful.
- A system is not a structure without dynamics.
- A system is not dynamical behavior without structural constraints and requirements.
- A system is not immune to its environment.
- The activity is the core building block of a service system model.
- Resources are objects with inseparable operant and operand properties.
- Money is one type of resource.
- A value proposition is an emergent outcome of a service.
- Reductionist modeling and the recognition of emergent phenomena are mutually supportive.
- Essential elements for building service system models include
    - Actor
    - Will
    - Value
    - Agent
    - Role
    - Resources
    - Access

  - Activity
  - Usage or yield
  - Authorization
  - Engagement
  - Engagement decision
  - Context
  - Journey

CHAPTER 4

Service Ecosystems

Having defined the atomic structure of service systems, we are now able to expand our model to the larger assemblage of structures that comprise a service system. Clearly, the activity element is too simple to capture multistage or coordinated parallel actions, but this simplicity advantageously allows the modeler to configure an endless variety of service systems and systems with unlimited complicatedness. We now turn our attention to the structure and function of complete service systems composed of the elements that we defined in the previous chapter.

A useful model of even common services reveals a surprising amount of complicatedness. Furthermore, the recognition of all actors and agents that take part in knowledge-based intelligent systems (KBIS) or knowledge-intensive business systems (KIBS) delineates a service system with numerous independent initiatives and motives in the execution of a service. For example, an airline reservation service is performed through the interaction of several airline data servers, a hosting site such as Orbitz or Travelocity, a traveler's personal schedule and travel requirements, advice from family members and friends, ground transportation web sites, online map services, and other information servers. We are led to define the features of the service system as a conglomeration of the service system elements. The term "ecosystem" has come into common use to identify such systems (Wieland et al. 2012). We are also indebted to the research of the Viable Systems Approach (VSA) (Golinelli 2010) for much of the system structure that is described in the following.

System Structure

Governance

Service scientists use the term "governance" to label the entire collection of rules, regulations, policies, and conventions that constrain and enable the service activities through social, governmental, and corporate institutions. The mechanisms of governance can be formal and informal, explicit and implicit. Governance can be implemented with a diverse set of codes such as

- The corporate policies that direct and limit the efforts of marketing personnel in providing customer service;
- The cultural norms that influence individuals in their interactions with online services;
- The sanitation laws that ensure the safety of hospitality services; and
- The protocols for requesting information in support of a service activity.

Agents must follow certain rules for their roles in contexts. Attorneys, physicians, professors, and other professionals are required by law to protect the privacy of their clients. Online services that accept credit card payments are required by financial accounting standards to ensure the security of credit card data. Healthcare services are expected by conventional industry protocols to share information about patients without regard to competitive positioning. Customers are expected to wait their turns in lines at all service facilities by cultural norms of fairness and respect. Rules for agent behavior abound and are crucial to explaining the performance of a service system.

Service activities must be controlled to remain within bounds dictated by sociogovernmental conventions, policies, and laws. Food service activities must adhere to standards established by sanitation laws, although these standards can vary substantially from country to country. Airline regulations require that flight attendants ensure that all passengers are securely seated prior to takeoff. Services provided by government agencies must be accessible to persons with handicaps. Modeling the rules that describe the resource transformations of service activities often incorporate governance requirements.

Resource usage must be feasible. The time required by a student to execute assignments must be within reasonable bounds according to a university's standards for student workload. The access of a product recommendation system to a client's personal big data must be fast enough for the system's algorithms to process the data quickly enough for a timely response. The modeler must be aware of the constraints on resource usage by service activities for each context as various governance modes can impose binding constraints on the usage.

To the modeler, governance is a set of rules to be incorporated in service activity transition laws. Although some rules can be global or institutional in scope, the modeler must be prepared to represent context-dependent governance rules bringing the modeling of the rules to the level of system elements. The good news is that governance, by definition, is codified and more precise than other aspects of human creative efforts. See Golinelli (2010) for a thorough explanation of the role of governance in service systems.

Categorical Values

Governance of service systems involves more than hierarchical decision making. Agents are governed by unspoken rules and guidance, as well as the expressed regulations of institutions. Categorical values is a term used to identify the dimensions of the ethical or prejudicial values that motivate or constrain agent decision making. The viewpoints, perspectives, prejudices, biases, opinions, standards, and world views of an actor are passed on to the agent in a service engagement. These guiding principles can be institutionally based or inherent in an individual. For the modeler, categorical values are murkier than governance rules because the former are usually unexpressed. Although they may be difficult to observe and measure, categorical values cannot be ignored by the modeler, as they can have a profound effect on agent actions.

Institutions

The term "institution" has many common meanings, and there are many kinds of institutions, but for the purpose of service system modeling, we will adopt a precise definition. See Vargo, Weiland, and Akaka (2015) for recent research on the definition of institution and its role in service systems. As with all of the specifications of the modeling framework for service systems that are presented in this book, we adopt here a definition that admits maximum robustness and parsimony of the model constructs. Our definition of institution requires only the most essential features of institutions, as they relate to service.

Definition: *An institution is defined by a governance and a set of categorical values.*

This rather broad definition allows many possible forms of institutions. Clearly, a company, university, provincial government, or religious congregation can be an institution. In these examples, the actors who participate in the institution are rather clearly defined. But an institution can exist without a persistent set of actors, agents, and resources. Our definition of institution admits organizations that can be somewhat amorphous such as the institutions of marriage, federalism, or human rights. We recognize and belong to many institutions in our daily lives without a clear idea of the governance and categorical values that define each of them. For the service system modeler, this vagueness presents a challenging indeterminacy in the behavior of actors and agents. In some service innovations, such as the design of a consulting agreement or a syllabus for a course, the parameters of governance and categorical values can be made very specific and measurable. In other cases, such as the implementation of a special interest group, a volunteer organization, an online collaboration site, a vacation tour group, and a software users group, the boundaries of governance and the shared categorical values are imprecise.

We can classify institutions along two dimensions. Each institution has its own governance, but the control by the governance can vary from rigorous to flexible. Each institution has categorical values that can range from precise to fuzzy. Figure 4.1 illustrates the two-dimensional classification scheme for institutions. Institutions that are highly flexible in governance and very imprecise in categorical values contain service systems that require evolvability in order to be effective. Such institutions may be highly inefficient. Institutions that are rigid in governance and precise in categorical values can be very efficient as long as the mission of the

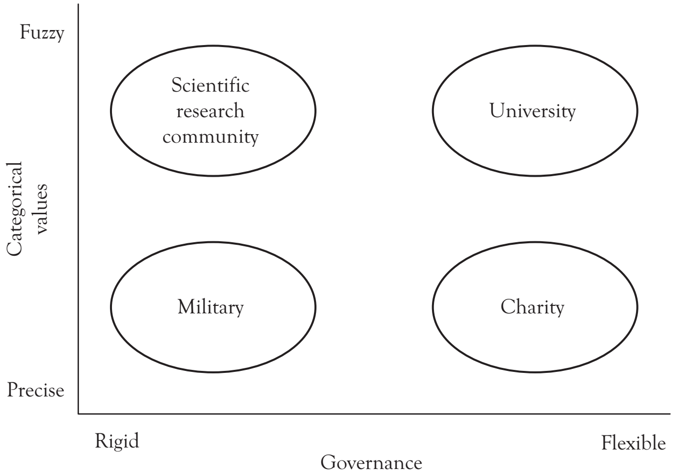


Figure 4.1 Institution classification

institution is relevant. If environmental conditions change, however, the effectiveness of such an institution can diminish rapidly.

Institutions promulgate ethics. Governance and categorical values can become codified formally or informally into a code of ethics. As the ethics of an institution become more explicit, they serve to delineate and bind the elements of an institution.

Our precise definition of institution in terms of our previously defined elements of governance and categorical values admits at once a clear distinguishing representation of institutions and a very broad interpretation of this construct. For example, a product brand is an institution. We have seen the impressive performance of brands such as Nike and Apple through the intense loyalty of their customer base who see themselves as belonging to a community of shared values and who consider themselves participants in a cocreative process with the companies as opposed to owners of a product that was supplied by the companies. The service system modeler can make effective use of the construct of an institution to identify the boundary-setting parameters of actors who are participants in the system.

Actors who seek service can usually choose among institutions and, during a service journey, switch from one institution to another. For

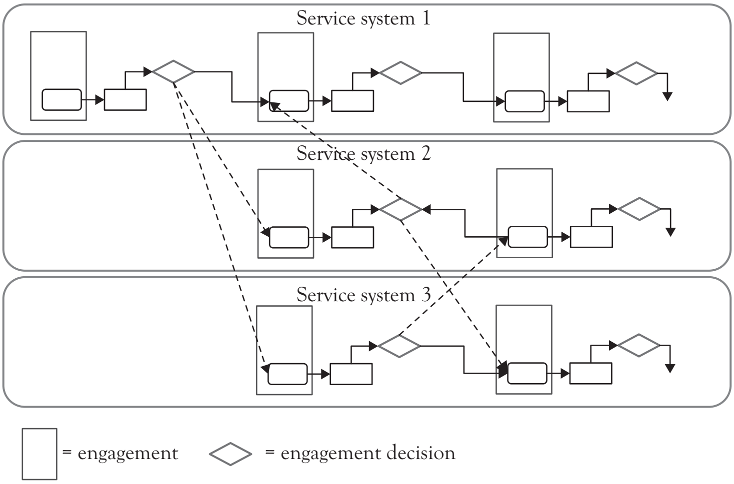


Figure 4.2 Multisystem service journeys

example, a customer of online retailers can easily switch from one retailer to another effectively leaving one institution and joining another, as each retailer imposes its own governance. Of course, this switching is a vexing problem for the providers of online service. Figure 4.2 shows how the engagement decision at each context of a service allows an agent to switch from one service system to another, allowing for multiple possible journeys. Institution switching can incur transactions costs through governance constraints and requirements.

Each service system then strives to retain the commitment of actors by ensuring that the perceived value achievement and the value potential of continued engagement is competitively high and sufficient to override the transactions costs of switching. A service system modeler is, therefore, charged with modeling the entire ecosystem of service institutions in the same domain of interest in order to compete.

Borders

Certainly, a service system with its assemblage of actors, agents, resources, and their interactions through activities and institutions constitutes a complicated system. System science is a well-developed field of study with origins traceable to the derivation of laws of thermodynamics in the 19th century. Applying the principles learned by studying systems of gas molecules in a closed container, the study of systems expanded into the modeling of biological systems, ecosystems, economies, social networks, and many other living and nonliving configurations of interacting members. To date, many researches in service science attach the nomenclature of physical systems (e.g., energy, matter, entropy) to service systems (Barile 2009; Golinelli 2010). Unfortunately, these representations lack the precision that allows useful modeling.

Service systems present a modeling challenge because they are open and amorphous. From the viewpoint of thermodynamics, an open system is defined as the one that exchanges energy and matter with its surrounding environment. Immediately, we have a problem in identifying the analog of energy and matter for service systems such as retailer websites, health care, education, and so on. We also have a problem in delineating the boundaries of such systems.

System boundaries are amorphous. People come and go in service systems such as online retailers, amusement parks, medical clinics, and universities, and all of these actors contribute to the cocreation of value with other actors. Then, where are the boundaries of the service system? Can we define a border that, more or less, corresponds to the boundaries of the system? Clearly, this border is constantly changing and the system model must recognize the nonstationarity of the system's scope. As a practical matter, a conventional system model will require the support of some approximating assumptions about system borders. A decomposition of the system into reasonably well-defined institutions as Figure 4.2 illustrates is one such approximation.

The Service Ecosystem

Human-centered service systems are the essence of daily life and economies. A city's viability depends on a coordinated execution of numerous interacting service systems. Public services such as traffic control, utility services (water, electricity, gas), law enforcement, environmental control and sanitation, fire suppression, emergency medical services, educational service, telephone, and Internet service not only share resources and information with each other but also with private service systems such as food service, legal service, and many others. In other words, the city is an open system of systems—an ecosystem. A casual observation of everyday life in any community of the modern world reveals similar complicated systems of service systems.

Evolving hierarchies of service systems can be modeled as ecosystems. Although the grand challenge of modeling an ecosystems as large as a city has been posited by several research communities in recent years, this challenge must be considered a big, *wicked* problem (Ng et al. 2012a). For pragmatic service system modeling with concepts and tools currently known, we can suggest the hierarchy of systems shown in Figure 4.3 as a framework for capturing the structure of service systems.

We can now see how the service systems that we modeled in Chapter 3 are subsystems of the service ecosystem. Consider the example of a couple of people performing an online selection of a restaurant for dinner. The actors, agents, and activities are subsystems of a context, as the two individuals engage a website for a particular restaurant and apply their knowledge of cuisine (operant resources), monetary, and time budgets (operand resources) to the activity of evaluating the menu that is posted on the website. Each context is a subsystem of a service system, as the couple decides to pursue the investigation of a particular restaurant

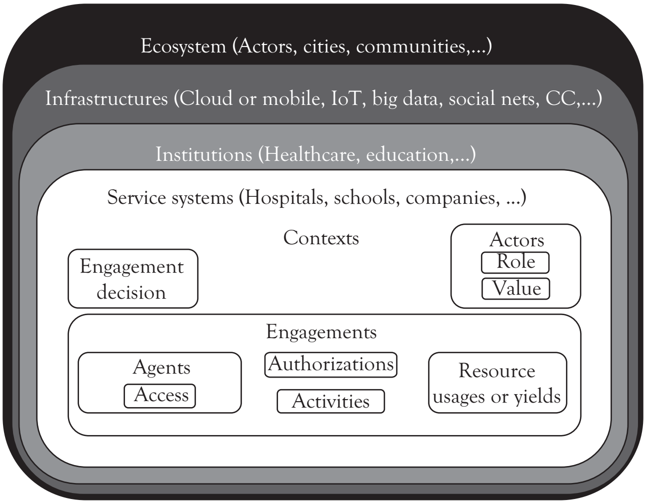


Figure 4.3 Hierarchy of service model constructs

further by clicking on other tabs or links on the restaurant's web page that provide information about location, parking, reservations, and so on.

Each service system is a subsystem of an institution, as the couple decides whether or not to redirect their investigation to websites of other restaurants in their community, all of whom cater to the type of clientele that the couple represent (categorical values) and follow a common set of rules for making reservations, sanitation, billing, and so on (governance).

Each institution is a subsystem of an ecosystem, as the couple integrates their restaurant search with traffic analysis from online traffic updates provided by local government, restaurant recommendations from social media as well as face-to-face conversations with friends, cuisine research from cookbooks (printed and online), weather reports from a television weather channel, and financial service in the form of an updated balance of their checking account.

Dissonance, Consonance, and Resonance

We now can recognize the importance of several macroscopic measures of service systems. Thermodynamics specifies macrosystem measures such as temperature, pressure, heat, and entropy. What measures are useful in describing service systems, institutions, or ecosystems? The VSA provides an answer by defining the performance measures of dissonance, consonance, and resonance (Barile 2009; Golinelli 2010). These measures derive directly from our model of service contexts and the effects of agent decision making in guiding the service journey.

In many service systems, such as KIBS and KBIS, the service journey reflects learning and adaptation by agents. As agents receive feedback from each activity engagement, they decide if and how they will reengage the service system. These decisions are made with imprecise understanding of the nature of ensuing contexts and uncertainty about the resources that will be made available to those activities. For example, the couple investigating restaurants engages a service activity every time they access a new web page. After viewing each web page, the couple has to make a decision with three options:

1. Continue to investigate the restaurant by clicking on other tabs or links on the restaurant's website.
2. Switch to another restaurant's website.
3. Declare the service complete, either with or without value creation, and terminate the search.

Each of these decisions reflects the knowledge that the couple glean from each engagement with a web page. Hence, the couple's journey through the service systems, illustrated in Figure 4.2, is determined by adapting engagement decisions, which in turn are determined by the knowledge produced by all previous engagements.

The VSA has identified three generic phases of learning and adaptation as dissonance, consonance, and resonance. This theory provides a more detailed view of the learning and adaptation that occurs in a service journey. Dissonance, consonance, and resonance (in that order) are valuable modeling concepts for learning and adaptation that fall between the many detailed and disparate theories of learning available from education research and the simplistic decision analysis of traditional operations research. Furthermore, these stages of learning and adaptation establish a sequence for contexts in a service journey that is valuable to the modeler in designing a viable order of activity engagements that cocreate the learning experience with the service agent.

Dissonance is the first stage of learning and adaptation in a service journey and the stage that is most likely to instigate a rejection of a service proposition. The word dissonance indicates that the integrations of resources by more than one agent in a service context are not consistent and their definitions of the activity requirements and outputs are not precise. A dissonant context can lead to a failure to create value-adding resource outputs and a rejection of future engagements in the service system. Although dissonance is undesirable, it is often inevitable in service because actors engage in service without sufficient prior knowledge. Even in the simple case of the couple investigating restaurants, initial contact with a restaurant's website may instigate confusion about the potential value of the site because of language barriers and a website layout that is not supportive of the users' purpose in viewing the site. An alternative approach to website design is to assume that on initial contact, a user's perceived utility of the web page is dissonant, requiring a homepage to be inquisitive of the user and determine the user's purpose and expertise before recommending options for further engagement.

Consonance is the second stage of the service journey which is marked by the service recipients and service providers achieving a common understanding of the service proposition. When consonance is reached, all agents that are integrating resources in the context have a common language and a common specification of the outcomes of the service activities being executed. However, these outcomes may not be value adding for some of the actors involved in the context. Therefore, consonance does not ensure the viability of a service.

Resonance is the third and final stage of a successful service journey which is marked by cocreation of value. When resonance is achieved, not only have all agents achieved a common understanding of the service activity but also the expected outcomes of the activity will produce value for all actors. In this stage, the service system is viable.

A service system designer should be aware of these stages as the design and development of the value proposition coevolves with the understanding of agents throughout the service journey. Achieving resonance requires adaptation by all actors, and in some cases, this adaptation may be beyond the capability or capacity of an actor. Therefore, achieving resonance may not be desirable for some service engagements. Service modelers should, therefore, consider the clientele for which a service is appropriate and beneficial to all parties.

Hierarchical Models

Conventional models of enterprises segregate functions into a hierarchy of strategic, tactical, operational, and technical levels that are integrated through a chain of command and feedback communications. Models represent systems from these different perspectives and reflect different decision-making levels in an enterprise.

Service systems are not as simple. Decision making in cocreative ecosystems is distributed both horizontally and vertically. Furthermore, lines of authority and governance structure are less distinct than in traditional views of business organizations. The hierarchy shown in Figure 4.3

is founded on the nesting of systems, networks within networks, and hypernetworks within hypernetworks, as opposed to the well-known tree diagram of organizational structures.

Summary

- Service is achieved through systems of systems.
- Governance is the compendium of regulations, decision rules, and codes that constrain and enable a service system.
- Categorical values are the ethics, priorities, positions, and biases that influence actors' behaviors and decisions.
- An institution is defined by a recognizable governance and categorical values.
- A service ecosystem consists of a nesting of infrastructures, institutions, service systems, and contexts.
- A viable service journey is typically a sequence of learning and adaptation through stages of dissonance, consonance, and resonance.

CHAPTER 5

Modeling Languages for Service Systems

Armed with the definitions and principles laid down in the previous three chapters, we are ready to fill a toolkit for modeling service systems. The natural extension of the context model of Figure 3.3 is the hypernetwork flow model of Figure 4.2. As long as we view service systems as mechanisms that transform resources through activities, then some kind of "stocks and flows" representation of the system seems inevitable. Consequently, network models of service system abound in the service-science literature. In this chapter, we survey and compare a compendium of these models with the aim of furnishing the reader well-rounded capabilities for constructing useful models of service systems.

Ontologies and Languages

In the remainder of this chapter, we review several of the most popular and useful network modeling methodologies for service systems. These methodologies can be called modeling languages, for each is defined with symbols, semantics, and syntax. Modeling languages enable the abstraction and encapsulation of system characteristics. Furthermore, many of these languages are graphical, allowing pictorial representations of service systems, which enhances the intuitive understanding and communicability of service-system models.

What is the lingua franca for modeling service systems? Certainly, a technical subject such as modeling systems should have a precise language with which modelers can communicate system designs without ambiguity or confusion. However, the answer to the question is, there is no universal standard. In recent years, there has been much discussion about and suggestions for the specification of an ontology for service and service systems (e.g., Mora et al. 2011; Petrie, Hochstein, and Genesereth 2011). To date, no widely accepted ontology has emerged and the definitions of service-system elements that we posited in earlier chapters is an attempt to lay some groundwork toward a service ontology that can be more or less uniformly accepted.

Standardization is vital to the usefulness of the models. The advantage of rigor in defining a modeling language is the precision of expression that this rigor enables. Modeling languages provide standardized specifications for drawing the diagrams of service networks and for imbuing the elements of these networks with properties. Standardization enables several key purposes of models:

- Modeling supports scientific endeavors and a model cannot be validated if it is not precisely defined.
- Modeling supports engineering endeavors and a model must be conveyable in clear, unambiguous terms to those who implement the model.
- Modeling supports managerial endeavors and a model must define specific, measurable, attainable, relevant, and timely (SMART) key performance indicators (KPIs) for evaluating system performance.

Standardization requires governance. The languages that emerge as standards within the scientific and engineering communities usually are given an official specification by some professional society. Of particular note in the domain of languages for process and systems modeling are the Object Management Group (OMG, www.omg.org) and Organization for the Advancement of Structured Information Standards (OASIS, https://www.oasis-open.org/). Such societies provide rigorous governance systems for the invaluable service of providing language specifications and for updating the specifications as needs change and new modeling initiatives arise. Through this governance the languages maintain their viability as communication tools among system designers.

The first step in establishing a language is the definition of symbols and semantics—an ontology (Mora et al. 2011). Each domain of science and mathematics has its origins in the form of precise definitions of the elements that form the domain of discourse. Formally, an ontology specifies an alphabet, words, and associations among words within this domain.

The new science of service lacks a single, comprehensive ontology. Several initiatives have been launched to provide this foundation, but the breadth of the subject and the novelty of the concepts of Service Dominant Logic (SDL) inhibit consensus on the content and structure of a universal language for service. Nevertheless, some elements of this language are granted widespread, if not universal, understanding, and these elements are sufficiently thorough in scope to support useful model building. This fact is the motive for this book. The reader can consider the terms and their definitions provided in Chapters 3 and 4 a naïve ontology for general-purpose model building. Languages grow and develop through use, and it is my hope that as practitioners and researchers produce a widening corpus of service-system models, the requirements for broadening and making this language more precise will emerge.

In the following, we review some significant initiatives at building ontologies for service or for certain aspects of service. Each one reflects a perspective or orientation about service and, as such, may not be appropriate for modeling all service systems. Nevertheless, the commonalities that one can find in these ontologies inspires hope for the evolution of a universal standard.

Resource-Event-Agent (REA)

The REA framework was initiated by McCarthy in 1982 (Geerts and McCarthy 2002; McCarthy 1982). REA was created to generalize the accounting standard of double-entry bookkeeping to cases of intangible resource exchanges. An event is a transaction between two parties in which resources are exchanged. The parties are called agents. REA is a rudimentary ontology that defines semantics for the exchange of tangible and intangible resources among agents in a service ecosystem. REA is used for economic analysis and lacks a proprietary graphical form (Schuster and Motal 2009).

Linked Service System for Unified Service Description Language Modeling Language (LSS-USDL)

LSS-USDL is an ontology that is based on a six-point star of interactions that take place in a service. USDL (http://linked-usdl.org/) is a modeling standard that was initiated in 2008 by the need to establish a unified framework for describing online and B2B service systems. Led by SAP, the thrust of the USDL development has been the precise definition of an ontology for software developers who construct the information technology (IT) components of service systems (Cardoso and Pedrinaci 2015). USDL has been updated to conform to the standards of Resource Description Framework (RDF) of the Semantic Web and Linked Data to capture cloud data resources in the design of service systems.

Clearly, LSS-USDL is a valuable ontology for developers of web-based services and service-level agreements. The specification of LSS-USDL is designed to be intelligible to this class of service modelers. Although the ontology defines semantics and object classes for many of the elements of service systems defined in Chapter 3, its specification is structured for IT applications and its broader adoption may be hindered by this orientation. More recent advances from SAP and its collaborators in developing a more general service ontology can be found in Ferrario et al. (2011).

Other Initiatives

Notable advanced and collaborative work on building ontologies for service are underway at the following institutions:

- Laboratory for Applied Ontology, Institute of Cognitive Sciences and Technologies, Trento, Italy.
- Department of Industrial and Business Information Systems, Information and Software Engineering Systems Group, Center for Telematics and Information Technology, University of Twente, Enschede, The Netherlands.
- Ontology and Conceptual Modeling Research Group, Federal University of Espirito Santo, Vitoria, ES, Brazil.

- The Adaptive Service Model© by the crowdsourcing initiative called *Taking Service Forward*, http://takingserviceforward.org/Taking_Service_Forward_-_Welcome.html.

Networks as Language Structures

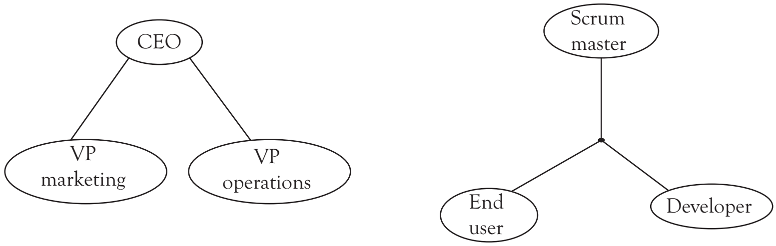
Structure and Evolution

Fundamentally, networks are models of systems that graphically depict the system relationships. We can define two categories of these relationships. The objects that make up a system can have a variety of structural relationships among them such as the governance that binds agents to actors, the access rights of agents for resources, and the communications channels that link actors to an engagement decision. A second class of relationships represent the transition laws of a system. A system evolves through time through transitions of the state of the system. In our structuring of service systems, the activities described in Chapter 3 perform this function. Hence, the relationships of activities to agents and resources enables a network representation of system evolution with an unlimited variety and dissemination of state-changing relationships.

Network models can be used to capture static structure of a systems elements or the evolution of the system over time under the influence of internal and external forces. However, the semantics of these two aspects of a system call for different representations. Hence, the many network languages for modeling systems provide different forms for these two complementary features of a system. Figures 3.1 to 3.3 and 4.2 are simple examples of network diagrams.

Graphs and Networks

To define a network, we first need to define a graph. A graph is a mathematical structure that consists of two kinds of objects—nodes and arcs. Conventionally, nodes are depicted by circles, ovals, or rectangles and arcs by lines or arrows. By connecting two nodes with an arc, we define a dyadic relationship between two entities. Expanding on this concept,



Graph of dyads

Hypergraph of a triad

Figure 5.1 Graphs

we can construct relationships among three or more objects as hypergraphs, shown in Figure 5.1. Fundamentally, graphs illustrate objects and the relations among them. Of course, there are mathematical representations of object relations in the form of order pairs of objects, ordered n-tuples of objects, and matrixes of relation valences. However, graphs are the most intuitive conveyance of relational information. For the service-system modeler, a graph is a natural framework for describing service systems and an indispensable worksheet for innovating a service system.

In service systems, typical binary relations include sequencing of activities, resource flows, state changes, communications, authorities, and so on. Nodes are usually used to represent actors, agents, resources, and activities.

We can view each node or arc as an object with properties and procedures. Some authors define a network as a type of graph in which the relations (arcs) have properties or valences (Wasserman and Faust 1994). However, in service systems, arcs representing relationships of system elements can be complicated enough to require a data set of properties of different data types. For example, a social network is a network of actors and the relationships among them, which can be quite complicated and require several data elements to capture their properties. See Wasserman and Faust (1994) for a thorough description of network models for social networks. Very often some of the metadata of a relation or entity is included in the network diagram alongside the arc or within the node, respectively.

Hypernetworks and Open Systems

Modeling subsystems and supersystems or multiple interacting systems requires a graphical representation known as a hypernetwork. By now, the examples of service systems described in the previous chapters should have convinced the reader that most service systems are quite complicated and that a model of a service system must be a model of systems of systems. Real-world service systems are never closed. Numerous external actors and agents are available to an actor who is engaged with a service system. The Internet is an ever-present provider of alternative services, knowledge bases to support service, and communication links to people who can join in the value-cocreation process. Even a customer in a restaurant can engage many other service networks through a smart phone in order to investigate the nutritional content of the meal on the menu, to compare the menu to meals offered by other restaurants, to acquire the advice of a friend about the quality of the food at the restaurant, and to share the dinner experience with a distant family member. In other words, service systems are networks of networks—a hypernetwork. See Chan and Hsu (2009). Figure 4.2 illustrates a hypernetwork.

In most cases, service systems are complicated enough to require a breakdown of systems into subsystems. In addition to the connections of one service system to others, within a service system, there are levels of detail that imply a hierarchy of subsystems. For example, a simple office visit to a doctor engages the service system of the office and its resources of receptionists, nurses, technicians, and doctors. Within the office, there are service subsystems of X-ray service, laboratory service for testing blood samples, nursing services for recording vital signs, physician services for diagnosis, and so on. The office service system of subsystems exists within the larger service system of health care at the regional or national level. This service supersystem involves insurance companies, government welfare agencies, the medical professional societies, and a network of hospitals and clinics. Therefore, in modeling a service network, we must consider both horizontal connections to other networks and nesting relationships of subsystems and supersystems. We will refer to the nesting of subsystems within systems as a hypernetwork hierarchy.

Graphical Modeling Languages

A Taxonomy

We will review a number of graphical modeling languages. For each language, the reader will be treated to a very brief overview as follows:

- The history of the language.
- Typical applications and the popularity of the language.
- Limitations of the language.
- Applicability to modeling service systems.

Our main interest is in the utility of these modeling techniques for representing the service systems as we have specified them in the previous chapters. Like all languages, modeling languages evolve over time and use and, in some cases, become arcane through diminishing use. Some of these languages are widely known and have long histories, and others are just out of the box, undergoing cautious scrutiny and modifications. Some of these languages were not invented specifically for modeling service systems, but they have features and structures that lend themselves well to the elements and hierarchy of elements of service systems that we have defined. As service science is a new science, service modelers are in the position to beg, borrow, steal, and innovate the tools that they need.

The hierarchical structure of the modeling elements shown in Figure 4.3 provides a standard for determining the applicability of each modeling technique to service systems. The definitions of the previous chapters established an aggregation or composition structure to the many model elements and collections of elements that are used to model a service system. For easy reference, this structure, which we shall call the Model Breakdown Structure, is shown in Table 5.1.

Value Network Analysis (VNA), e^3

VNA and e^3 are languages for coarse descriptions of value cocreation accomplished through defining the roles and exchanges of tangible and intangible resources among roles of actors or agents in a service ecosystem. VNA was created by Verna Allee (2009). VNA is a

Table 5.1 Model breakdown structure

Ecosystem
-Infrastructure
--Institution
---Governance
---Categorical value
---Journey
---Service system
----Context
----Actor
-----Role
-----Will
-----Value
----Engagement decision
----Engagement
------Authorization
------Activity
------Resources usage or yield
------Agent
-------Access

rudimentary form of expressing relationships among roles in a service ecosystem by identifying the tangible and intangible resource exchanges that take place between roles (http://www.valuenetworks.com/home.html).

e^3 was created by Jaap Gordijn and Hans Akkermans (2001). e^3 has been expanded into a family of modeling ontological frameworks that includes e^3-value, e^3-control, e^3-strategy, e^3-alignment, e^3-service, and e^3-boardroom (http://e3value.few.vu.nl/). Like REA, e^3 modeling standards assume that every transaction between two parties must involve some kind of reciprocity—a true exchange. e^3 was created for mapping e-commerce value chains and is a graphical modeling language. Notice that the unit of study for these modeling languages is the ecosystem and that the unit of analysis is the exchange. These modeling standards do not reveal operational details about value cocreation or resource exchange. Both of these modeling standards decompose the service into binary exchanges between pairs of actors, agents, or roles (Schuster and Motal 2009).

Integration Definition (IDEF)

One of the oldest network models for service systems is known as IDEF (http://www.idef.com/). IDEF was created through a research project of the U.S. Air Force (USAF) in 1970s called Integrated Computer Aided Manufacturing. The initiative to create the IDEF modeling language was the need to design efficiency, safety, responsiveness, and readiness into the manufacturing and logistics support services of the Air Force's mission. This analysis necessarily involved modeling processes other than conventional manufacturing processes.

IDEF is a family of languages. Over more than three decades, the IDEF initiative succeeded in defining 16 IDEF languages in order to support modeling at different levels of a system hierarchy and for different application domains. Table 5.2 shows the list of these languages

Table 5.2 IDEF language hierarchy

IDEF language	Service system model elements
IDEF0	Journey modeling
IDEF1	Information modeling
IDEF1X	Data modeling
IDEF2	Simulation modeling—replaced by IDEF 3
IDEF3	Process modeling
IDEF4	OOP
IDEF5	Ontology
IDEF6	Design
IDEF7	Information system
IDEF8	Human–computer interaction
IDEF9	Business constraints
IDEF10	System architecture
IDEF11	Information artifacts
IDEF12	Organizational design
IDEF13	Three-schema architecture
IDEF14	Network

and their application domains identified in terms of the elements of service-system models. The USAF program developed IDEF0, IDEF1, IDEF2, IDEF1X, IDEF3, IDEF4, and IDEF5. Like all languages, IDEF has undergone evolutionary refinements, enhancement, extensions, and partial obsolescence over the decades of its use. Although there are more than 16 languages in the IDEF family, most of these are not actively supported and some have been made obsolete by others. See Noran (2004) for a brief history of the IDEF evolution. Currently, the Knowledge Based Systems, Inc. maintains a standard for IDEF0, IDEF1, IDEF1X, IDEF3, IDEF4, and IDEF5 (www.idef.com), which are the IDEF languages that are in common use.

Of all of the IDEF modeling languages, the IDEF0 language is best suited to modeling service systems at the level of activities and journeys. The IDEF0 language is deceptively simple. The objects of the IDEF0 model are nothing more than rectangles and arrows. The rectangles represent activities and the arrows represent resource inputs and outputs of activities. The descriptive power of IDEF0 is made precise by segregating inputs and outputs into four categories. Material resource inputs and outputs are allowed to connect to an activity box only on the sides. Information inputs and outputs are allowed to connect to an activity box only through the bottom of the box. Authorizations for an activity are resource inputs and outputs that are allowed to connect only to the top of an activity box. Figure 5.2 shows the IDEF0 diagram for the entire service system for making a flight reservation.

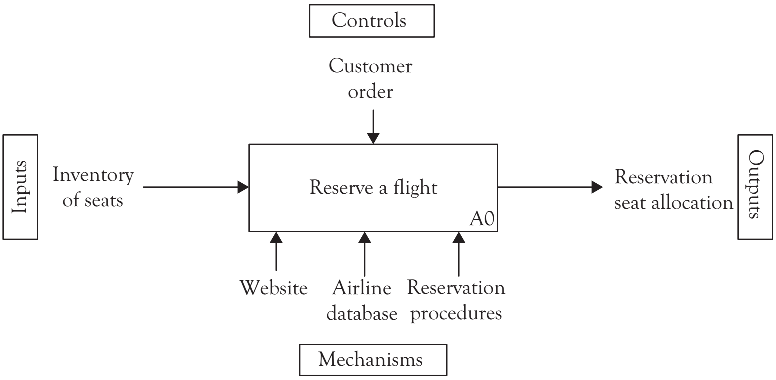


Figure 5.2 Flight reservation system—IDEF0 level 0

Another key feature of IDEF0 is its layering. Through layering, IDEF is ideally suited to modeling a hierarchy of subsystems. Each activity is given an identification in the form of the letter "A" followed by a number. The diagramming begins with a single activity box labeled A0 that represents the entire service system. On another sheet, this system is subdivided into two to six subsystems. These subsystems are labeled A11, A12, …, A16, the first digit indicating level 1 of the layering. Continuing in this way, a service system can be decomposed into as many subsystems as the modeler requires in order to capture the process details that are the subject of innovation. Through such simplicity, IDEF0 can be learned quickly and a modeler can apply this method at any level of detail in analyzing a service system. Figure 5.3 shows the first level of the decomposition of the flight reservation system described in Figure 5.2.

There is a rich history of IDEF applications. An interesting example of a service system is the process of product design. Over the past few decades, product design and development has emerged as a service that is critical to global competitiveness. An interesting example of the use of IDEF0 modeling to analyze an existing product design system and to design an improved product design system can be found in the PhD dissertation of Regan (1997), which applied IDEF0 to the innovative improvement of the annual updating of clothing designs by apparel manufacturers. Lest any reader suspect that IDEF is too simple to capture

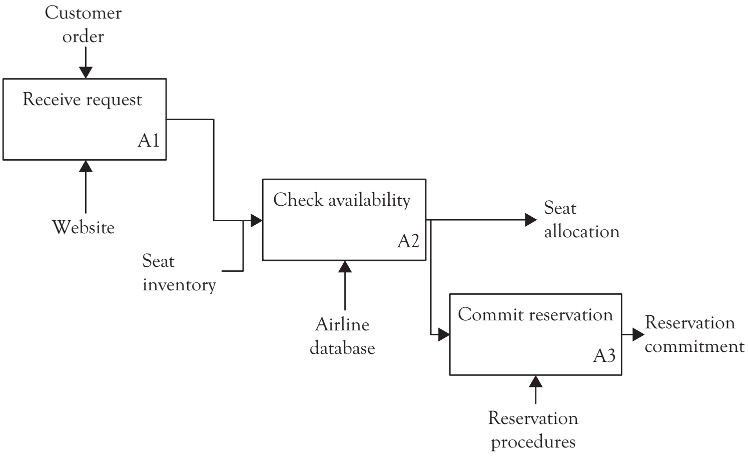


Figure 5.3 IDEF0 diagram of a flight reservation—level 1

intricate process details, this example, produced by product designers in four apparel companies, was carried to five layers and required the definition of more than 200 activities.

Unified Modeling Language (UML)

UML is a language specification published by the OMG (OMG, 2014a). OMG was founded in 1989 as a worldwide, nonprofit consortium of companies, government agencies, academics, and other individuals who share an interest in standardizing practices and languages that are used in software and systems development. OMG has contributed mightily to the development of object-oriented programming (OOP) and other standards. Once established as a reputable developer and custodian of standards for OOP, OMG expanded its scope to related fields and the processes that interface with and use IT resources. The result is collection of dozens of rigorous language specifications. Among these specifications, UML is perhaps the most widely known and used.

UML is a modeling system that is well known to information system developers, which has been developed and promoted by the OMG since 1997 (OMG 2014a). The avalanche of development of large, complicated, and increasingly interconnected software packages in the 1990s instigated a design doctrine for software products that are scalable, extensible, flexibly configurable, easily debugged, and resilient. Software modularization led to the common practice of OOP, which is codified by UML. Although its motivation was software development, object-oriented modeling is a useful framework for modeling all systems. Furthermore, the UML standard serves as a precise and generic language for this modeling. Even if the service-system modeler chooses not to apply UML, a review of the UML is instructive for understanding the structure of all bonafide modeling languages that are reviewed in this chapter.

UML provides a nine-level hierarchical system of network models for system design, management, and operation (Rumbaugh, Jacobson, and Booch 1999). It is clear from the UML standards that UML was designed with systems engineering in mind and OOP in particular. Although both IDEF and UML have features for capturing an OOP-based system design, system developers tend to favor UML for this purpose, as the emergence

Table 5.3 UML language hierarchy

UML model level	Service system model elements
Class	All elements
Object	Instances of all elements
Use case	Access, activity
Sequence	Journey
Collaboration	Institutions
State	Journey
Activity	Context
Component	Activity
Deployment	Infrastructure

of OOP in the 1980s strongly motivated the design of the UML family of languages (Noran 2004). Table 5.3 shows the entire UML hierarchy of languages with their application domains interpreted for the context of service-system modeling.

OOP constructs are very robust. As Table 5.3 indicates, the class model can capture every element of a service system. In general, a UML class is a set of objects, each of which is characterized by a set of properties or attributes and a set of procedures or functions, through which all the objects perform state-changing actions. Furthermore, associations defined among classes allow the modeler to represent any relationships (hierarchical, structural, causal, informational, etc.) between classes. In consequence of these broad definitions, the Class is the building block of object-oriented software and also serves as a flexible conceptual model of the components of any system. Accordingly, we could use the UML Class model to specify all of the elements of a service system. A full description of UML classes, associations, and other UML model components is far beyond the scope of this book. The reader is referred to the large body of literature on OOP. A good starting point is the OMG website, www.omg.org.

Figure 5.4 illustrates an abbreviated class diagram for a service-system model of an airline reservation system. Figure 5.5 shows a portion of a UML sequence diagram for the activity of placing a reservation through a website.

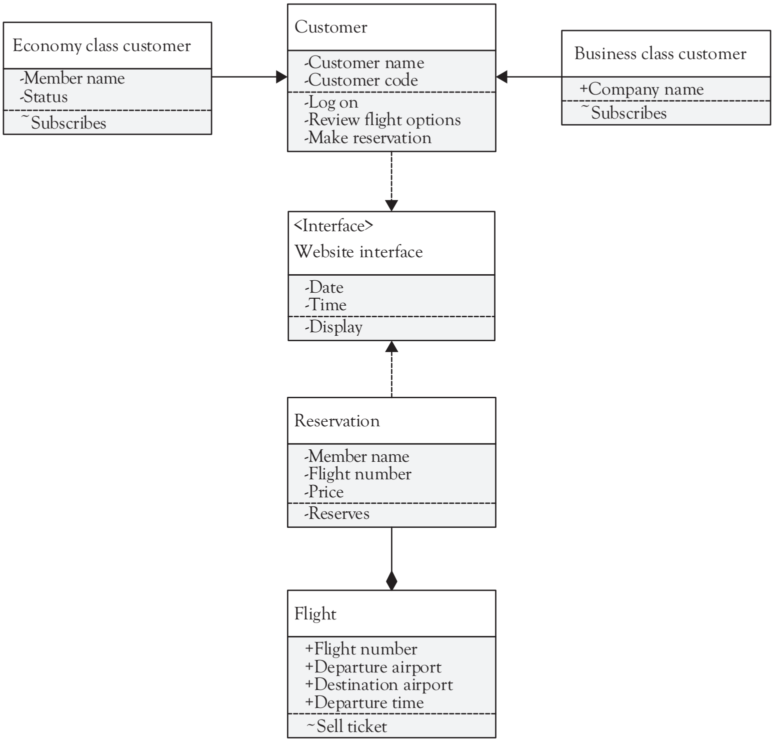


Figure 5.4 UML Class diagram for airline reservation

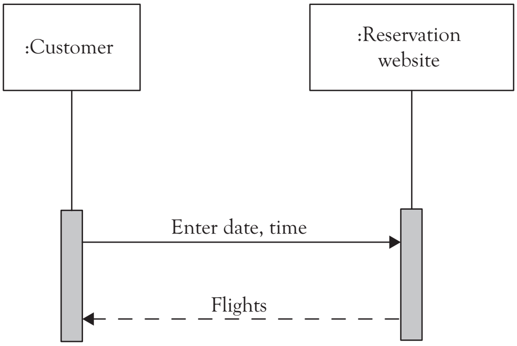


Figure 5.5 UML sequence diagram

Service Blueprint

One of the most familiar network models is the service blueprint. See Shostack (1984), Bitner, Ostrom, and Morgan (2008). This framework

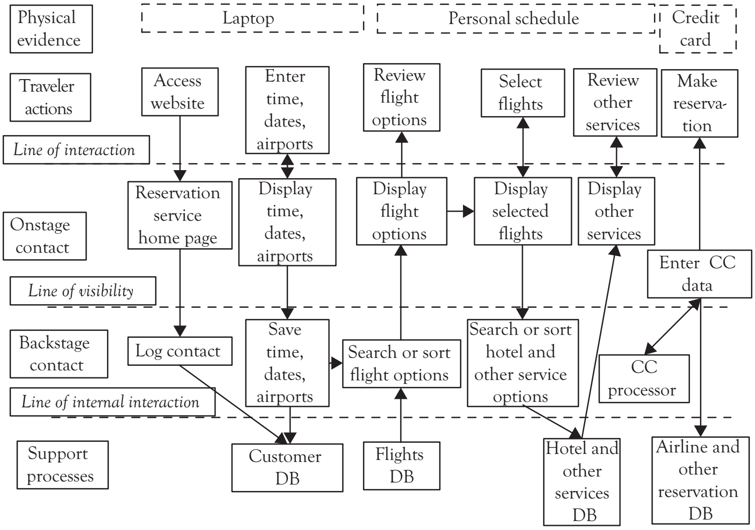


Figure 5.6 Service blueprint

for describing service systems consists of only a handful of semantic elements and syntactical rules. Hence, the service blueprint is very robust and can be adapted to many service systems. Figure 5.6 shows an example of a service blueprint for making a flight reservation. All of the elements of this language are shown in the figure. The blueprint shows the journey of a service through a series of activities. A service is described in terms of activities represented by the rectangles in the diagram. The inputs of resources to and the outputs of resources from activities are denoted by the arrows in the diagram. A key feature of the service blueprint is the segregation of service activities into four categories represented by four horizontal bands or "swimlanes" in the diagram. Swimlanes are convenient graphical devices to identify different sets of agents and the boundaries between them. Each swimlane is populated by activities performed by a well-defined set of agents.

The swimlanes from top to bottom are

- Physical evidence—the tangible objects through which the client of the service engages the service providers;
- Customer actions—the activities, sequenced from left to right in the blueprint, through which the client engages the service;

- Line of interaction—the boundary between the client agent and the client-facing service agents;
- Onstage contact—the service activities performed by the client-facing service agents (human and nonhuman);
- Line of visibility—the boundary between the client-facing service agents and the backstage service agents;
- Backstage contact—the activities performed by the first line of service agents not in direct contact with the client agent;
- Line of internal interaction—the boundary between the backstage agents and the supporting agents; and
- Support processes—the activities performed by agents, internal and external to the service provider, that provide support resources to the backstage agents.

A service blueprint is one of the most basic network models for service innovation at an operational level. The service blueprint clearly allows the modeling of the journey through activities that comprise a service and, through the use of supporting swimlanes, has the capability to capture a hypernetwork of service systems. The advantage of swimlanes defined in terms of individual agents or sets of agents is the clear allocation of responsibilities for each activity in the service system, making the service blueprint a useful device for designing a service system. One can easily visualize a service provider, such as an airline in the example shown in Figure 5.6, using the service blueprint as a worksheet for examining a service system for opportunities for preventing failures and for improving efficiency by reengineering activities and resource flows.

Business Process Model and Notation (BPMN)

BPMN is a modeling language that has been managed by OMG since 2005. Version 2, released in 2011, contains features that enable it to capture the details of a service activity, contexts, resources, and communications (OMG 2011). BPMN was motivated by the need to map business processes, which grew out of the numerous process improvement initiatives such as TQM, JIT, 6-sigma, BPR, and all of the versions of Lean. For many years, the approach to IT development was to consider the installation of technology a project in its own right, managed by IT professionals. The abysmal failure rate of these projects over several decades (more than 50 percent according to most studies) revealed a fundamental myopia of the technology focus—the purpose of technology is process improvement. This viewpoint of the role of technology engendered a sea change in project management. Instead of scoping such projects with the objectives of IT *installations* followed by user training, the new view scopes projects in terms of *implementation* of process improvements. Process improvement begins and ends with people and processes. Technology enables process improvement, and technology design and development should be driven by the demands of process improvement. In other words, technology is subordinate to process. When process owners took over the management of process improvement projects, they discovered that the tools of UML were too technical and IT oriented for intuitive representations of business processes and the BPMN initiative was born.

BPMN is intuitive and transparent to anyone who wishes to describe the procedures, interactions, resources, and information flows of common experiences in business systems. This language, like IDEF, is more flexible and robust than the traditional flow charting methods that have been used to model manufacturing processes for the past century. BPMN was designed with this generality in mind. The applications of this language have consequently been quite varied. One can apply BPMN to a manufacturing supply chain as easily as one can apply it to the operations of a service such as banking, health care, emergency service, and any service in which people, equipment, and information are engaged in processes according to some structured sequence or in adaptation to changing contexts.

The relative simplicity and intuitive design of BPMN also limit its applicability to modeling the details of data structures and function calls of an information system. BPMN should be considered a language for describing how systems of people and resources make use of information and IT as opposed to describing the inner workings of the IT. Therefore, the role of BPMN in system design is at a higher level than that of UML.

BPMN is ideally suited for describing the activities and journeys of a service system. Figure 2.1 shows a BPMN diagram for the activities of a healthcare clinic. Note that a basic feature of a BPMN model is the set of "swimlanes" that are used to assign service activities (called processes in the BPMN standard) to particular agents or actors. Swimlanes can also be defined as the service systems within which various agents or resources engage in service activities. Resources in the form of information or data are explicitly represented in BPMN. Furthermore, messages of various types that can be used to initiate, terminate, or control engagements of a context are given specific representations in the BPMN language. Subsystems can be modeled in BPMN through activity groups.

The sequencing of engagements is also explicitly shown in a BPMN diagram. Sequencing is differentiated from resource flows into and out of activities. Engagement decisions can be explicitly represented as well through the contingency nodes of the BPMN diagram. In this way, the BPMN language can convey both structure and journey of a service system.

Business Process Execution Language (BPEL)

BPEL is language standard that is specified and maintained by the OASIS group since 2003. Web services motivated the development of BPEL and the language specification is based on the procedures for transferring data, information, and commands between a user and a website. As web interfaces are ubiquitous in the service economy, BPEL has come to be viewed as an essential language for modeling business processes. BPEL4People is an extension of BPEL to enable the modeling of human interactions and human processes with web-based systems. Hence, a marriage of BPMN and BPEL seemed natural and several attempts to find the common features of these languages and to integrate the two have been made with somewhat limited success. However, the strong connection between BPEL and BPMN and their concomitant use in the design of web-enabled service systems motivates the inclusion of BPEL in this compendium of modeling languages.

BPEL is not a graphical language. BPEL is an *XML standard.* Extensible Markup Language (XML) is widely used and relatively simple structure for entering coded markups to text in order to imbue machine-interpretable semantics to the text. For example, the marked up phrase <title>Modeling Service Systems</title> identifies the content

"Modeling Service Systems" as the title of a document. The simplicity of markup languages makes them very robust, and there are numerous standardized XML schemas for different application domains. The BPEL standard was developed as one of these schemas.

BPEL text can serve as a framework for developing computer code and as a modeling tool to sort out the procedural details of a web-based service. With the addition of BPEL4People, this framework can be extended to the service activities carried out by agents of human actors in a web-enabled service system. However, the capabilities of BPMN for the noncomputerized aspects of a service system as well as BPMN's limited capability to capture the overall functionality of a web interface secure a firm adjunct position to BPEL in service-system modeling.

Similar to UML class diagrams and sequence diagrams, BPEL captures service processes at a higher level of detail than a BPMN diagram can. Therefore, BPEL and BPMN are complementary languages in describing the specific details of the transfer of information between humans and computer systems in a web-mediated service system.

Service Value Network (SVN)

A SVN can be used to show a graphical depiction of a *mashup* (van Dinther et al. 2011). The explosion of service innovations in recent years has, in the main, been due to the configurations of applications and application interfaces that enable online and mobile applications. The creators of these "apps" are able to design, build, and deploy sophisticated and innovative service systems by applying a methodology that has been known to the manufacturing world for several decades—modular design. If a product or service can be designed as a configuration of standard modules, then a wide variety of offerings can be made with a relatively small number of standard modules. An obvious example of modular design is the laptop computer which, with a few dozen components, can be configured in literally thousands of products according to individual customer requirements and preferences. If each component of a product or service can be assigned to one of a set of available modules and all of these available modules for one component are compatible with all of the available modules for each of the other components, then the variety of offerings can be maximized for the given set of modules.

The application of this concept of modular design is now evident in the design of online and mobile apps which are often called mashups because these service systems are cobbled together from catalogs of standard application interfaces for performing basic functions such as recording a user's registration, checking passwords, receiving credit card payments, interfacing with other systems such a web browsers and e-mail, and so on. The use of the SVN language is motivated by the importance of these forms of service-system design, which is driving the *API economy* (Collins and Sisk 2015). An SVN is defined as a service that is created through the configuration of such modules that are available through registered and vetted providers of applications and application interfaces. In order to model the construction of this kind of configuration, a simple, graphical modeling language for SVN was created. Figure 5.7 illustrates the SVN for the case of airline reservation service.

The SVN shows the entire set of available applications for each stage of the service with the large ovals within which specific attributes (a_j^i) of the application are listed. The third-party providers of these applications are shown and connected to their offered apps by dotted lines. Solid arrows indicate the feasible configurations of the applications that the service-system designer can implement. The cost of invoking an application can be entered above the arrow that enters the application into the configuration. Although it is desirable for all applications to be compatible with

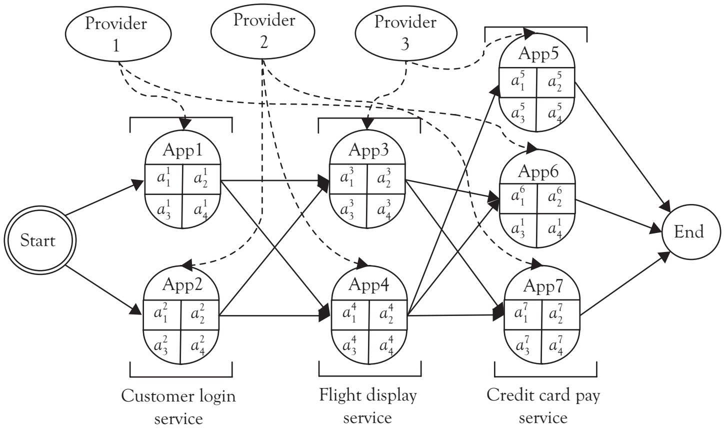


Figure 5.7 SVN for airline reservation

all other applications, there can be cases of incompatibility that reduce the number of configuration option, shown in Figure 5.7, by the absence of an arrow from App3 to App5. The SVN portrays to the system designer all of the configuration options.

Yet Another Workflow Language (YAWL)

YAWL was created in 2002 by Wil van der Aalst of Eindhoven University and Arthur ter Hofstede of Queensland University (ter Hofstede et al. 2010). YAWL is a business process modeling language describing the sequences of tasks, or workflows, that are required to accomplish a service objective. Workflows are the software applications and manual tasks that support business processes. YAWL is an open-source language standard that is now supported by the YAWL Foundation (yawlfoundation.org). A windows-based model editor is available to build graphical models for workflows and to specify data requirements of the workflow.

As the name of this language standard implies, YAWL is designed for describing workflows, which positions the standard at the activity level of a service system. As such, YAWL is a close competitor to BPMN and we will describe YAWL by comparing and contrasting it to BPMN. In fact, BPMN2YAWL is a translator that converts BPMN diagrams to YAWL diagrams. The pedigree of YAWL is a mathematical modeling framework known as colored Petri nets. YAWL extends this framework to a broader and more intuitive modeling context.

The YAWL framework is quite simple and transparent. Under the YAWL framework, a system is modeled as set of nets, which may be nested in a hierarchical structure of subnets. A net is similar to a swimlane in BPMN. Each net has a starting condition and an ending condition. A workflow is contained within a net. A workflow consist of tasks (think of a task as a service activity). Tasks can be manual or automated. Tasks are connected to one another by flows, which serve to sequence tasks in a network. Splits and joins allow the network of tasks to branch and converge flows. YAWL also supports the definition of roles, which can be assigned to personnel who are considered resources.

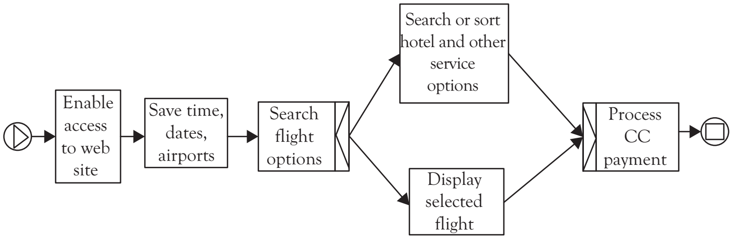
YAWL also describes data requirements of a business process. Data are transferred from the net to the task and from the task to the net. Data are not transferred from task to task. This approach supports modularity.

"Decomposition" of tasks is the interface that allows the task to receive or send data to external functions or databases. A decomposition specifies the data definitions of any data that are sent to or received from the external environment. Data definitions are made through XML schema.

Although YAWL is designed for describing workflow patterns that, in general, can be any sequenced network of tasks, YAWL is decidedly IT oriented. The language was conceived by and for system developers. Unlike BPEL, which for some modeling initiatives might be considered a competing standard, YAWL has a graphical interface. YAWL is simpler than BPMN in terms of the alphabet of symbols and elemental constructs required, but its creators claim that through its simplicity, there is robustness and that YAWL has broader expressive capability. See the example of workflows in Figure 5.8.

Computational and Configurable Service System (C^2S^2)

Weske (2007) extended the YAWL constructs to a more service-oriented framework by incorporating human interactions and knowledge transfers in the business process. Qiu (2009, 2014) described the application of this extension to a network model of service. The motivating perspective of these extensions is that of business process management (BPM) applied to value-cocreating, human-centered systems. Qiu (2009) labeled his network model C^2S^2, which stands for "computational and configurable service system." Under this structure, a graph of conditions (C) and tasks (T) connected by flows (F) form a directed graph. Flows can go from condition to task, from task to condition, and from task to task. The network structure is hierarchical and the nodes, flows, communications,



Reservation service net

Figure 5.8 YAWL workflow

and activities are well-defined object classes. YAWL symbols and the pi-calculus are used to formalize the structure.

Value Delivery Modeling Language (VDML)

The reader may detect an unsettling goods-dominant perspective in YAWL, C^2S^2, BPMN, BPEL, UML, and IDEF. The heritage of engineering practice since the industrial revolution through the information age has guided design practice in all its forms with the mantra "necessity is the mother of invention." Consequently, we tend to think that designing a service system begins with a recognition of a need by a market. What follows is a well-known sequence of activities that comprises the design process. Table 5.4 demonstrates the parallel approaches of conventional product design and its legacy in conventional service design.

In service design, and in particular in the design of information systems to support service, we can see this pattern. However, the modern perspective of service calls for a paradigm shift in design theory and practice. In this text, we began a journey of learning about service by recognizing a fundamental, salient feature of all service as cocreation of value. This cocreation begins with the design of a service. We have established the foundation of service systems on relational constructs instead of on transactional constructs, the latter being enabled by the former. Let us not underestimate the importance of this statement. Its implications are dramatic:

- Markets do not exist, they are cocreated.
- Value propositions cannot be predefined, such as a blueprint for a product.
- Value propositions emerge from a service.

Table 5.4 Design process synopsis

Goods-dominant design process	Inherited service design process
Identify market need	Identify market need
Specify product performance	Define the value proposition
Design the product	Design the customer experience
Design the production process	BPM for the service system
Release product to market	Engage service recipients

Hence, a modeling language that can express truly collaborative and unpredictable design and execution is needed. VDML is a bold initiative toward fulfilling this need.

VDML is an interesting recent development by OMG. VDML is distinguished from other business modeling languages in several aspects that are essential to representing service systems.

- Tangible and intangible resource uses.
- Capacitated and consumable resources.
- Flexible, multiparty collaborations for coproduction.
- Distinction between ownership of capabilities and access to capabilities.
- Agile, context-dependent activities.
- Explicit representation of value and value creation from resources.
- Distinction between actors and roles.
- Explicit representation of value propositions.
- Shared capabilities for cocreative activities.

The creators of VDML define value as “a measurable factor of benefit delivered to a recipient in association with a deliverable” (OMG 2014b). Although VDML has some elements that are similar to those of BPMN (e.g., swimlanes for roles, activities, flows), the reader is cautioned not to consider VDML as a special application of BPMN. On the contrary, VDML was created for the purpose of modeling the aspects of a service system above the operational level. Specifically, VDML provides semantics and syntax for service aspects such as capabilities (operant resources of SDL), value propositions, value derived from resource delivery, and the construction of unconventional and dynamic organization structures for cocreating value. VDML fills a gap in the model structures that can be used for service systems.

VDML can trace its pedigree to VNA, *e*^{3} value modeling and REA analysis. Note that the unit of study for these modeling languages is the ecosystem and that the unit of analysis is the exchange. These legacy modeling standards do not reveal operational details about value cocreation or resource exchange. By contrast, VDML extends the analysis of value cocreation to an operational level and is a language that is designed to describe activities that engage the capabilities of multiple participants to cocreate value through collaborations.

VDML defines numerous classes of objects that express the structure and dynamics of a service system. The hierarchy of classes in VDML begins with two broad categories: model elements for defining the structure of a service and measurable elements for defining particular implementations of the service. The reader who has no experience with VDML will be impressed by the close correspondence of the names of the elements of VDML to the elements of service systems that were defined in Chapter 3. VDML Object Classes are defined hierarchically as follows.

1. VdmlElement
    a. Attribute
    b. Annotation
    c. MeasuredCharacteristic
    d. Operand
    e. Expression
    f. CalendarService
    g. PortDelegation
        i. InputDelegation
        ii. OutputDelegation
    h. BusinessItemLibrary
    i. BusinessItemLibraryElement
        i. BusinessItemDefinition
        ii. BusinessItemCategory
    j. Assignment
    k. ReleaseControl
    l. ValueDeliveryModel
    m. AnalysisContext
        i. Scenario
        ii. DelegationContext
    n. ValueDefinition
    o. ValueLibrary
    p. ValueCategory

    q. CapabilityLibrary
        i. Capability
        ii. CapabilityDependency
    r. PracticeLibrary
    s. PracticeDefinition
    t. PracticeCategory
    u. RoleLibrary
    v. RoleDefinition
    w. RoleCategory
2. MeasurableElement
    a. BusinessItem
    b. CapabilityOffer
    c. DeliverableFlow
    d. PortContainer
        i. Collaboration
            1. BusinessNetwork
            2. Community
            3. OrgUnit
            4. CapabilityMethod
        ii. Activity
        iii. Store
            1. Pool
    e. Port
        i. InputPort
        ii. OutputPort
    f. Role
        i. Participant
        ii. Party
        iii. Member
        iv. Position
        v. Performer
    g. ResourceUse
    h. ValueProposition
    i. ValueElement
        i. ValueAdd
        ii. ValuePropositionComponent

VDML expresses key features of cocreative systems that are not found in other modeling languages. VDML semantics is unique in capturing most of the features of a service system. The relationships of the classes of objects listed previously further illustrate the power of VDML to express the structure and dynamics of a service system. Several of the core relationships are as follows:

- Activities are performed by Roles within a Collaboration.
- Collaborations can be OrgUnits, Communities, BusinessUnits, or CapabilityMethods.
- Actors are Participants in a Collaboration by being assigned Roles within specified Activities.
- Collaborations invoke Activities.
- Activities apply Capabilities to procedures that transform input ResourceUses to output ResourceUses.
- In order to model flexible, context-dependent execution of Activities, the VDML model delegates the Activity to a DelegationContext that is invoked by a Collaboration.

VDML is integrated with the Structured Metrics Metamodel (SMM), another standardized language sponsored by OMG for rigorously defining measurements from observations.

VDML models service systems through the use of eight types of diagrams:

- Role Collaboration
- ValueProposition Exchange
- Activity Network
- Collaboration Structure
- Capability Library
- Capability Heatmap
- Capability Management
- Measurement Dependency

Figures 5.9, 5.10, and 5.11 are examples of a Role Collaboration Diagram, Value Proposition Exchange, and an Activity Network, respectively.

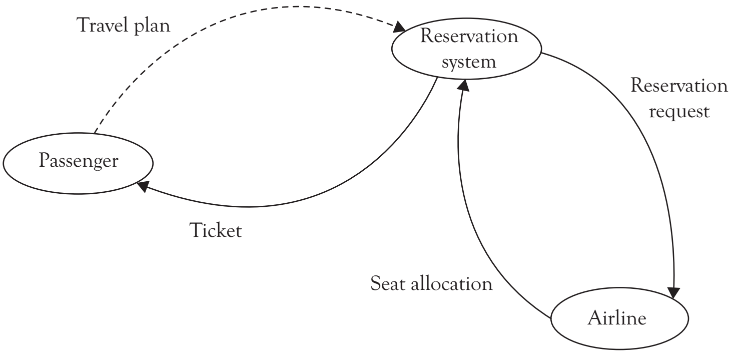


Figure 5.9 VDML role collaboration diagram

VDML is the only modeling language that has been developed specifically for modeling cocreative systems at both operational and tactical levels. The robustness of VDML for modeling service systems is evident in the inclusion of most of the basic elements of service system defined in Chapter 3 in the basic constructs of VDML. VDML is a new standard that will be developed further in the coming years by the service modeling community and holds great promise to become the standard language for modeling service systems.

*i**

*i** (http://www.cs.toronto.edu/km/istar/), championed by Eric Yu at the University of Toronto (Yu et al. 2010), enables modeling of social interactions and actor intention. The *i** standard originated with the realization that information systems designed to support increasingly sophisticated and complex service cannot be adequately designed from the goods-dominant perspective of traditional systems design. Mechanistic views of service systems consisting of entities and their relationships, activities and their inputs and outputs, or constraints and their parameters fall short of capturing the intentions and beliefs that activate the pursuit of value. Furthermore, intentions and beliefs are stimulated and modulated by social interactions within institutions. Therefore, it is impossible to model cocreative systems without imposing a superstructure of the sociality of intention over any model of behavior, interaction, or activity.

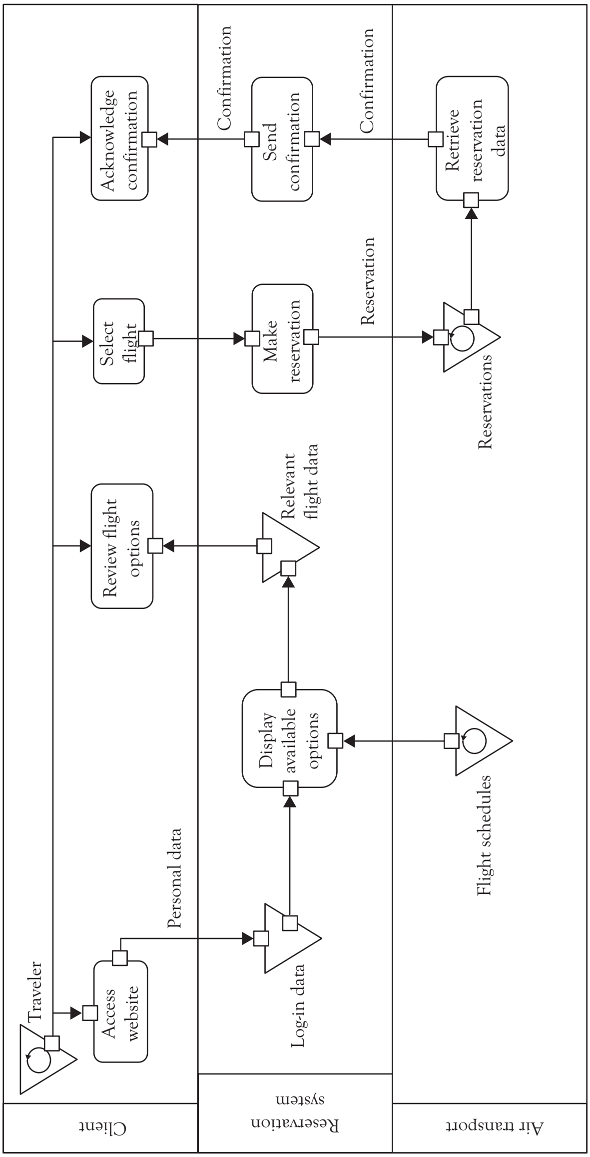


Figure 5.10 **VDML activity network**

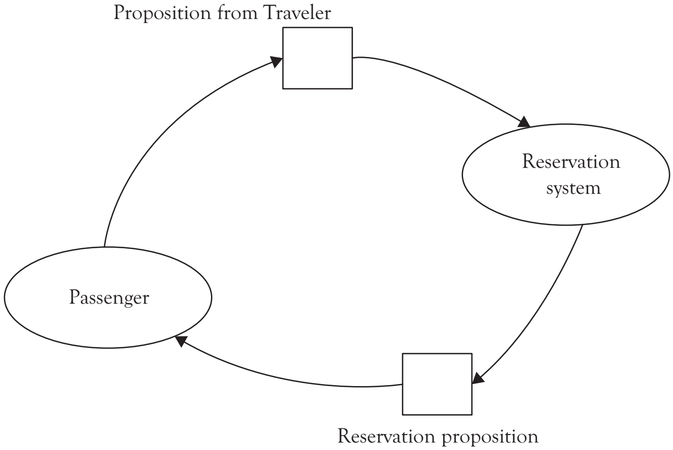


Figure 5.11 VDML value proposition

Dramatically departing from convention, the creators if *i** assert that goal-oriented modeling must precede solution-oriented modeling in the design of a system.

The main model construct in the *i** standard is the actor—human or machine, individual or group. Two fundamental premises about actors highlight the reformist standpoint of the innovators of *i**.

- Actors are autonomous—actions are not completely knowable or controllable.
- Actors are intentional, intentionality is distributed through social networks.

*i** models social interactions through networks of dependencies. A dependency is defined as a binary relation in terms of three elements:

- Dependum—the object that is exchanged in a binary social interaction.
- Depender—the actor that depends on Dependee for the Dependum.
- Dependee—the actor that provides the Dependum to the Depender.

Four types of dependencies are defined in terms of the type of Dependum that is involved in the relation.

- Goal dependency—Dependum is an aspiration or assertion that the Depender wants the Dependee to achieve or prove, without specifying how to achieve or prove it.
- Task dependency—Dependum is an activity that the Depender wants the Dependee to execute according to the Depender's instructions or parameters.
- Resource dependency—the Dependum is a resource.
- Softgoal dependency—Dependum is a nonfunctional requirement that may have multiple dimensions or imprecise specification.

The *i** modeling language is graphical. The following is a list of the object classes that make up that language and for which there are graphical icons.

- Actor—a conceptual entity that possesses intentionality.
- Agent—the physical enactment of an actor.
- Role—the logical enactment of an agent in a context.
- Position—a set of roles played by an agent.
- Goal—a KPI that represents a high-level aspiration of an actor.
- Task—an activity that can produce outcomes which support the attainment of goals.
- Resource—any knowledge element, datum, communication, or physical object that is used in the execution of a task.
- Softgoal—a multifaceted perceived benefit of an actor such as quality or comfort.
- Belief—a categorical value; not a goal, but a guidance in setting goals.
- Boundary—actors operate within institutional boundaries and context-dependent intentional boundaries.
- Links—in addition to the dependencies, there are four other kinds of relations among model elements.

- Dependency link—identifies a Depender–Dependee relationship in terms of a Dependum.
- Decomposition link—identifies task to subtask relationships.
- Means-end link—connects a task to a goal.
- Contribution link—the support of a softgoal by a task.
- Actor association link—identifies social relations among actors.

With these well-defined modeling constructs, the system designers can create two graphical models that formalize the social network, intentions, and possible combinations of tasks and resources that can support goal attainment. Note that the ambition of the *i** modeler is to identify and describe possible system designs that support goal attainment as opposed to designing a system that produces prespecified and well-defined requirements. The difference between these modeling objectives is significant, as it highlights the departure of the *i** approach to systems design from convention and the necessity of a social modeling approach to describing human-centered, context-dependent service systems.

The Strategic Dependency (SD) model is a dependency map that describes the interdependence of intentionality. The SD map displays dependencies only as the example in Figure 5.12 illustrates. A Strategic Rationale (SR) model displays all links. Furthermore, SR accepts multiple paths to the solution to a problem as opposed to prescribing a single final design of a system.

*i** inspired an extension called User Requirements Notation (URN), which has been adopted as a standard by the International Telecommunications Union (http://www.itu.int/ITU-T/aap/AAPRecDetails.aspx?AAPSeqNo=1806). Another extension of *i** is Value-Cocreation Modeling (VCM) created by Lessard (2015), which combines elements of Business Intelligence Modeling (BIM) (Horkoff et al. 2012), VNA (Allee 2009), and *i** in order to model the particular value creation modes of knowledge-intensive business service (KIBS) and knowledge-based intelligent service (KBIS). Although the modeling elements are drawn from other language standards, their combination enables a specification of two key processes to these kinds of service defined by Lessard as

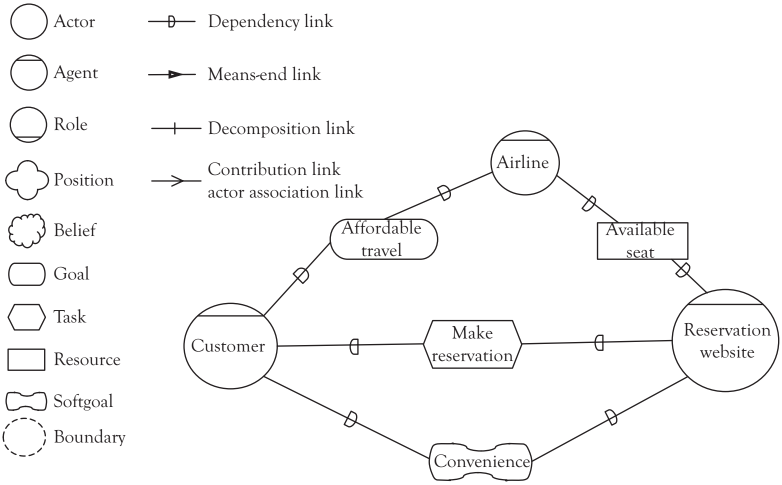


Figure 5.12 i* SD model

alignment and integraton (reminiscent of the concepts of consonance and resonance of the Viable Systems Approach (Barile 2009).

*i** and its derivatives represent an emerging new orientation of system modeling that views the system journey determined by the value-seeking intentions of actors as opposed to engineered processes that propose to deliver value. This orientation has the potential to illuminate and extend the now-popular agile approach to requirements generation and software development. Through this orientation, software itself, is being viewed as a social system—distributed, evolving, encapsulated, and associated.

Summary—Model Synopsis

Model breakdown structure	**LSS-USDL**	**BPEL**	**IDEF**	**UML**	**YAWL**	**Service Blueprint**	**BPMN**	***i****, **VCM**	**VDML**
Ecosystem									
Infrastructure									Y
Institutions									Y
Governance									
Categorical values								Y	
Journey								Y	Y
Service systems					Y	Y	Y	Y	Y
Contexts					Y	Y	Y		Y
Actor							Y	Y	Y
Roles							Y	Y	Y
Will								Y	
Value								Y	Y
Engagement decision									Y
Engagements	Y	Y	Y	Y	Y	Y	Y	Y	Y
Authorization	Y	Y	Y			Y	Y	Y	Y
Activity	Y	Y	Y	Y	Y	Y	Y	Y	Y
Resources U or Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Agent	Y	Y	Y	Y	Y	Y	Y	Y	Y
Access	Y	Y	Y	Y	Y	Y	Y	Y	Y

Y = Yes. The model element has an explicit representation in the framework.

CHAPTER 6

Decision Making

In this chapter, we discuss the *process* of decision *making*. The investigation of service systems in the previous chapters has revealed the essential role of decision-making activities in determining the journey of a service. In Chapter 3, we mentioned decision making as a type of activity within a service system. Specifically, engagement decisions direct the course of each agent that participates in a service. In fact, the defining characteristic of service, cocreation of value, demands agent decision making during the service. Therefore, in this chapter, we turn our attention to the study of decision making, and in the next chapter, we learn to create models of decision making. Such models will be necessary components of any model of a service system. In what follows, we will refer to the decision maker as the decision-making agent (DMA), as this is the entity in a service system that makes the engagement decisions.

What Is a Decision?

First, we need to define the word "decision." As an educator in the field of decision analysis who has taught decision modeling to many audiences—undergraduate, graduate, and experienced professionals—I have always been amazed at the widespread ignorance of the essential nature of decisions. Even among senior executives who supposedly base their careers on their decision-making powers, I have witnessed an inability to define the scope and trade-offs of a decision in a coherent manner. Therefore, at the risk of appearing to state the obvious, I must ensure that the reader understands two fundamental characteristics of decisions from which we can begin a discussion of decision making and decision modeling.

A decision requires a set of alternatives, with a corresponding set of performance measures, of each alternative. Alternatives can also be thought of as options, choices, or available actions. Performance measures evaluate consequences, outcomes, and ramifications of each alternative. For example, a person who engages the service of an online airline reservation system is faced with a decision—which flight to select. The alternatives are given by the list of available flights from the starting location to the destination. The performance measures include ticket price, convenience of the time of departure, convenience of the time of arrival, the discomfort of long layovers, and the quality of service of the airline. For each potential flight selection, the traveler (the DMA in this case) considers and trades off all of these performance measures. What is being measured is the performance of the DMA's proposed alternative.

In every decision, there are factors beyond the control of the DMA, which also influence the performance measures. The cost of the flight to the airline, the number of seats on each plane, the distances between cities, the demand for connecting flights, and so on, all influence the performance measures of this decision and must be considered in any model of the decision. These causal factors that are not within the alternatives that the DMA is empowered to select are called parameters of the decision.

In a nutshell, a decision is based on the cause–effect relationship between two sets of causative factors and the set of evaluative measures that the DMA uses in order to judge the desirability of each alternative. The causative factors are divided into a set of controllable factors that constitute the decision alternatives and a set of uncontrollable factors are called parameters. An alternative is chosen by the DMA. Parameters must be measured, estimated, or forecasted. The evaluative measures are called performance measures because they quantify the "performance" of each decision alternative. Figure 6.1 illustrates the fundamental structure of a decision.

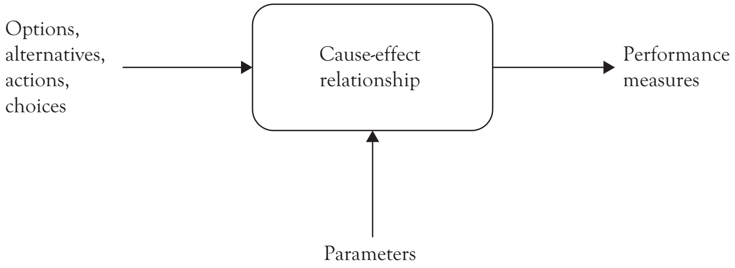


Figure 6.1 Decision structure

Rethinking: Fast Versus Slow Thinking

Nobel Prize winner Daniel Kahneman categorizes human decision making in two modes—fast thinking and slow thinking (Kahneman 2011). Fast thinking is the application of decision rules or heuristics to decisions, which enable quick, automatic decisions and require the decision to fit a preconceived pattern that the decision maker has experienced before.

Slow thinking occurs when the decision maker discovers that the proposed engagement does not conform to a known pattern. In this case, the DMA is forced to construct a new model of the decision—a learning process that typically stresses the patience and the ability of most people. Consequently, this process is often terminated prematurely, which leads to engagement rejection. The service system designer must understand this process and provide the appropriate learning experiences to engage agents at each stage of the slow thinking process.

When we recognize fast thinking in decision making, we usually overlook the fact that every heuristic, decision rule, programmed response, or habitual action has its origins in a model-building exercise through which the DMA studied the cause–effect relationship to the extent that acceptably good solutions could be derived, codified, and memorized. Nobody is born with an understanding that leaving a car unlocked can lead to theft, making a flight reservation later than two months before the flight will lead to higher ticket prices, or even placing a hand on a stove can lead to pain. We have to learn the decision rules that can be used to guide our actions automatically in certain contexts, which raises another poorly understood fact about decision making—learning is accomplished by model building. With some reflection, one can group one's knowledge into two categories: (1) facts and data and (2) concepts. Every concept is expressible as a relation among facts or data or both, much like the cause–effect relation that forms the basis of every decision. Therefore, conceptual knowledge takes the form of models. We can conclude that fast and slow thinking are not two disjoint modes of cognitive function but rather two levels of the learning process.

When a DMA models a novel context and pursues this effort to the point of identifying acceptable solutions for any configuration of parameters that specify the context, the DMA will derive a set of decision rules or heuristics for responding to each context. Once these lessons are learned, the shortcut from context identification to action can be programmed and automated. The underlying model by which the lesson was learned can be filed away or forgotten until the DMA is challenged by a new variation in the context, new desires or expectations for performance, or new options for the decision.

In the hypervariable world of IT-enabled service systems, which respond to a hypervariety of contexts, the DMA encounters incessant demands for learning new decision rules for new contexts. Furthermore, tolerable lead times for making a decision can be as short as a few seconds. Service systems are now tasked with modeling learning and decision making at lightning speed—slow thinking must be done fast! Consequently, decision *modeling* must be built into the decision *making* in every smart service system.

The Role of Decision Models

Decision making inevitably requires the decision maker to "construct," formally or informally, a model of the cause–effect relation between alternatives and performance measures. For an engagement decision, this model must be based on a model of the transformation of input resources to output resources of the proposed service activity. Most everyday decisions are made through informal, even subliminal, decision models. However, explicit, model-based decision support for decision making in business, engineering, and government has been under development and deployment since World War II and is now routinely applied at all levels of organizations and even within mobile apps for personal use. Whether a decision is made with the help of a computerized decision support system that optimizes the selection of alternatives against the trade-offs inherent in the performance measures or through an intuitive, heuristic decision rule, all rational or somewhat rational decision making involves an understanding of the cause–effect relationship from alternatives and parameters to the performance measures that are relevant to the decision maker.

Humankind is on the verge of a revolution in model-based decision making that will infiltrate every aspect of every person's life—smart service systems. Not only is decision making an integral part of service systems, but in the emerging generation of smart service systems, intelligent decision supports will be integral components of every service system. Understanding the nature of decision making has never been more important to a system designer.

Once a model of the cause–effect relationship exists, the decision maker can pursue a desirable decision by defining the decision *criteria*, defined as the conditions that the DMA places on the performance measures in order to specify what constitutes a feasible and optimal decision. In setting the criteria, the DMA is, in effect, defining the structure of *value*. For example, two different users of an airline reservation system may define a feasible, optimal solution according to the following two different criteria, even though they view the decision in terms of the same cause–effect model.

DMA #1 Criteria set: Choose a flight that

Minimizes price
Subject to the following constraints
  time of departure no later than 10 a.m.
  time of arrival no later than 10 p.m.
  layovers no more than three hours
  quality of service of the airline at least four out of five on satisfaction ratings

DMA #2 Criteria set: Choose a flight that

Minimizes layover time
Subject to the following constraints
  time of departure no later than 10 a.m.
  time of arrival no later than 10 p.m.
  price is no more than $500
  quality of service of the airline at least four out of five on satisfaction ratings

Clearly, the setting of criteria is a subjective, context-dependent, and value-laden task, which a well-designed service system should enable. Sounds simple enough? In this chapter and the next, we will unveil the less-than-obvious complications of this decision modeling challenge. Chapter 7 expands on decision modeling for the advanced reader. For now, we step back from the details of the decision model and discuss the process of decision making, within which the decision model is embedded.

Rethinking: Complexity Versus Complicatedness

Complex has become an overused adjective for describing cocreative contexts. Certainly, the engagement decisions and activities that direct a service journey are difficult to model because they are subjective, contextual, and novel to the DMA. But writing off the possibility of understanding these contexts with the label *complex*, as some researchers do, precludes the benefits of many effective analytical techniques for describing service contexts. If we are to meet this modeling challenge, we should demystify as much as possible the nature of service contexts. A valuable contribution of the Viable Systems Approach (VSA) research in this regard is a distinction between the words *complexity* and *complicatedness*, as they apply to decision making and service contexts (Barile 2009; Golinelli 2010).

Returning to the cause–effect relationship that forms the core of a decision and a decision model, real decision environments often introduce several issues that make achieving a thorough understanding of this relationship an overwhelming challenge. Many decisions, even common, everyday decisions, can involve hundreds or thousands of alternatives and a similar number of performance measures and parameters. Although these decisions may be precisely formulated, their analysis and optimization are intractable without computerized, model-based support. For example, every day electric service grids must make the dispatching decision of all of the available generation units in the grid for providing service over each of the 24 hours of the next day. The performance measures for this decision include the satisfaction of forecasted demand over each of the 24 hours in the planning horizon, the capacity constraints on each generation unit, the capacity constraints on transmission lines, the ramping constraints (limits on the rate of change of a generator's output), the cost to deliver power to each retail bus, and other physical constraints on the power flow in the grid. This decision involves literally thousands of variables, parameters, and constraints. Nevertheless, this problem is solved routinely every day by grid managers all over the world. Such a problem earns the label *complicated* because the problem is big. But the problem is well formulated because all of the dimension of the problem are well defined and measurable. All of the formulas that determine the values of the performance measures for each potential solution are known and precisely specified. Hence, the problem is complicated but not complex.

Another issue that arises in decision making is uncertainty. In the case of the electric service dispatching decision, forecasted demand for electric power, like all forecasted demand, is not known with certainty. We live in an uncertain world, and we must make decisions that affect our futures in this uncertain world. It is easy to spot uncertainty in typical engagement decisions within service systems. Travelers engaging a reservation service may not know in advance exactly when they need to be at their destination or what the quality level of each airline is. Students engaging an educational service do not know in advance how much effort a particular course will require. Patients engaging a health-care service may not know whether or not they can afford the treatments or medications that may be prescribed. Hence, parameters of most real-world decisions are not known with certainty.

How do we make plans under conditions of uncertainty? Some would say that there is no point to planning when one cannot predict the outcomes of those plans. This hypocritical or ignorant view of decision making belies the fact that everyone does make decisions in consideration of the uncertainty in the outcomes of those decisions. The answer to the question is risk analysis. Whenever a decision involves parameters that are not known with certainty, such as forecasted demand, the rational decision maker estimates the risk to the performance measures that this uncertainty engenders (Badinelli 2010). Then, this risk is traded off against other performance measures in choosing a course of action. In general, risk is mitigated in three ways: buffering, contingency planning, and hedging. In other words, uncertainty in decision making is routinely handled with well-established mechanisms which add dimensions to the performance measures and, in an analytical approach, require the estimation of probability distributions. Therefore, uncertainty and risk make a decision problem complicated but not complex.

Besides uncertainty, there is another form of indeterminacy in the cause–effect relation that governs decisions. Imprecision is different from uncertainty. Uncertainty exists when we cannot accurately measure the value of a well-defined quantity, such as future demand or the time that we need to arrive at a destination. Imprecision exists when we cannot specify the quantity that we wish to measure. Uncertainty measures inaccuracy; vagueness and ambiguity measure imprecision. For example, the quality of an airline could require the specification of several, poorly defined dimensions of customer service that are not well understood by the traveler who must choose an airline for an upcoming trip. Although the traveler may assert that quality of service is a highly important performance measure of the engagement decision, the traveler may be unable to precisely define the dimensions of this measure. This imprecision makes the problem complex. Note that a problem does not have to be big in order to be complex.

The inability of a decision maker to define the alternatives, parameters, or performance measures of a decision can be compounded by an inability to precisely define the cause–effect relations among these dimensions. For example, a patient may not know whether or not time spent viewing a website about the patient's condition will have any effect on the success of treatment. In this case, the cause–effect relationship between service engagement and valued outcomes is not precisely known. Finally, we need to recognize that the most important performance measure of an engagement decision, value, is the most imprecisely known dimension of the decision. Therefore, service engagement decisions are fundamentally complex.

A modeler of a DMA's engagement decision needs to distinguish complexity from complicatedness because *complicatedness* will succumb to conventional, well-designed decision support systems. However, *complex* features of an engagement decision impose a different kind of indeterminacy to the journey of a service. Complexity arises from vagueness and ambiguity in the specification of the proposed service engagement. True complexity is the mark of a decision that is not completely formulated in the mind of the decision maker. Fuzzy modeling techniques can be used to provide analytical support to such decisions but this form of modeling is starkly different from the modeling of uncertainty (Badinelli 2012).

By distinguishing the nature of uncertainty from the nature of vagueness, we clarify in fundamental ways the form of decision making that takes place in any given context. The complexity and complicatedness of a decision reflect two different aspects of a DMA's state of understanding and awareness of the service, which evolve in the course of successive engagements. Consequently, a well-designed service system is adaptive to an agent's state of understanding and facilitates the development of that understanding through flexible and robust engagement alternatives which enable *learning* and *adaptation*. VSA research provides valuable insights into this learning process.

Abductive, Inductive, Deductive Reasoning

The phases of learning and adaptation that were defined as dissonance, consonance, and resonance in Chapter 4 can be modeled more explicitly in terms of the slow decision making process of an agent in each of these phases. The first stage of decision-making is called the abductive stage. A DMA facing a new decision begins with a complex problem, and perhaps complicated problem, that requires some guessing at the alternatives and performance measures of the decision. The main challenge faced by the decision maker in this stage is posed by the complexity of the problem rather than its complicatedness. The problem may be large or small in scope, but before the DMA can contemplate the difficulties of finding a solution from among a number of alternatives that trade off a number of performance measures, the DMA must first be able to define the performance measures and options. Abductive reasoning is the most peculiar and unpredictable form of human reasoning. Some refer to abductive reasoning as "thinking out of the box." Abductive reasoning enables a DMA to consider cause–effect relations that the DMA has never experienced or witnessed before. Such reasoning can also lead to hypotheses that can be tested through the ensuing stages of learning.

Certainly, biases, experience, culture, institutional standards, and cognitive patterns weigh heavily on the path and outcomes of abductive learning. Defined as *categorical values* and *interpretive schema* by the

VSA doctrine, these features imbue the abductive learning environment with unique contexts. A well-designed service system should be malleable with respect to these context-specific influences on the direction of the abductive learning experience and should skillfully integrate knowledge resources from subject matter experts in a way that illuminates the nature of the proposed service engagement to the DMA who might be hamstrung by categorical values and interpretive schema.

Let's return to a simple example of a service system. Suppose an individual experiences lower back pain, and now this individual plays the role of a DMA by engaging online service systems to find a solution to this problem. Initially, the DMA does not have any idea about the cause of the pain or the possible treatments for it. Accessing one of the numerous medical websites, the patient questions whether or not the time invested in engaging the website will result in knowledge that creates value. At this stage of the service, the DMA does not have a precise definition of the alternatives that are available, the performance measures that are relevant, or the parameters than may be necessary to understanding the relationship between treatment options and outcomes. Unfortunately, many medical websites immediately launch the user into medical jargon and cursory diagnosis which stymies that DMA's need for a basic understanding of the nature of the problem. The DMA and the website are in a state of dissonance, which demands abductive reasoning (Barile 2009).

The second stage of decision making is called the inductive stage. Once the DMA has made preliminary investigations of the problem, a *framework* for the decision model emerges. At this stage of the service, the DMA is prepared to propose hypotheses about the cause–effect relationship between the symptoms and the condition for a diagnosis and, for a particular condition, between the treatment alternatives and the performance outcomes. The alternatives and performance measures have been identified and well defined. The nature of the effect of alternatives on performance is also defined precisely and needs to be validated. If the DMA with lower back pain has persevered through the abductive stage, the patient can now hypothesize that the diagnosis could be one of four possibilities: a slipped disc, a herniated disc, a cracked disc, or osteoporosis. Furthermore, the DMA has learned that potential treatments for these conditions include bed rest, nonsteroidal anti-inflammatory drugs, acupuncture, physical therapy, and surgery. Validating the diagnosis based on the DMA's symptoms and building an understanding of the likely treatment outcomes for each treatment and diagnosis is a model-building exercise by the DMA that requires inductive thinking—formulating hypotheses and validating them based on sample data, which may be anecdotal, personal history, or acquired from external sources. At this stage of learning, the DMA has achieved consonance with the website that has enabled the learning process (Barile 2009).

The third stage of decision making is called the deductive stage. After a model of the decision is fully specified, the DMA can deduce outcomes of alternative actions and evaluate the relative desirability of different alternatives. At this stage of the service journey, the DMA can enter the known symptoms to the diagnosis model and determine exactly what the condition is. Then, knowing in precise terms the nature and severity of the condition, the DMA can evaluate the performance of outcomes of every treatment alternative and the risks associated with it. Finally, the DMA can select the most appropriate treatment. The DMA in this stage of the service journey is thinking deductively, and the website has achieved resonance with the DMA (Barile 2009).

Online services and mobile apps experience massive rejection by DMAs. Users of these service systems flippantly transfer to other websites by making engagement or disengagement decisions based on cursory examination of the site's offering. Service system designers often overlook or underestimate the hurdle that service offerings must overcome in order to retain a DMA's attention and commitment through the dissonance stage. The same criticism can be made of many human-centered service systems that engage a service participant in a cocreative relationship such as student–teacher, physician–patient, lawyer–client, consultant–client, and so on. In all such service systems, there is often an initial "disconnect" between the two participants that is never resolved. Here the failure to recognize the dissonance that exists owing to the need for abductive learning thwarts the successful execution of the service.

The VSA concepts of dissonance, consonance, and resonance with the associated abductive, inductive, and deductive forms of decision support expose essential requirements of a service system and the necessary sequencing of those requirements. Service system designers should become well-versed in these VSA concepts and consider structuring service systems with user options for abductive or dissonant, inductive or consonant, and deductive or resonant engagements.

Hierarchical and Networked Decision Making

An enterprise, such as a firm or an individual's long-term educational program, is managed through a large number of interacting decisions. In an ideal world (perhaps we should call it a utopia), all decisions that guide an organization such as a firm would be made through one, giant, coordinated decision engine that optimizes the entire set of strategic, tactical, and operational plans—a truly global optimization. For example, a typical manufacturing company must decide, on an hourly interval, which production jobs will be performed on each work station within each work center, within each shop, and which personnel will perform these operations. The desired solution will meet demand at minimum total cost. In order to find a solution to this problem that is optimal, or even just pretty good, the company must forecast its requirements, capacities, and resource availabilities for several days or weeks into the future. Hence, the production scheduling problem involves thousands of variables and thousands of constraints. Putting aside complexity, the complicatedness of such a problem precludes this comprehensive solution. Consequently, organizations since the beginning of human experience have adopted structured approaches to decision making, which traditionally fell into hierarchical forms. It is no accident that organizational structures have conformed to this same hierarchy.

A hierarchical decision framework is fundamentally a system structure that positions "small" decisions subordinate to "big" decisions. A hierarchy of decision making is the only practical framework for achieving a reasonable degree of consistency and coordination among these decisions. Through the hierarchy a detailed plan can be derived from a rough-cut aggregate plan. The example of manufacturing planning described earlier illustrates a classic hierarchical decision-making structure.

In order to ensure that the *chain of command* in hierarchical decision systems is maintained, the decision governance must incorporate aggregation and disaggregation methods. A typical manufacturing firm generates an aggregate sales-and-operations plan with a monthly interval and a 12-month horizon that determines the aggregate values of inventory levels, overtime budgets, production volumes, and so on. The solution to this aggregate problem, in the form of the chosen values for these decision variables, become parameters on the lower level decision of setting the master production schedule. The master production schedule is typically a weekly plan for the production of final assemblies. This plan must conform to the budgets and targets set by the aggregate plan. Once the master schedule is determined, its solution becomes parameters that constrain the lower-level decisions of setting the production and purchasing plans for all subassemblies, components, and raw materials. From the weekly production plans, daily schedules can be determined for each work station. As the decision making moves down the hierarchy, the granularity of the decisions increase and the decision intervals decrease. Eventually, the firm determines the detailed schedule that it seeks without encountering a decision that overwhelms the personnel and information systems.

Familiar examples of hierarchical decision structures are found in government, industry, and the military. One only has to look at the organizational structure of an enterprise to see the hierarchical breakdown of decision making. Hierarchical structures tend to be rigid and suffer from top-down propagation of errors. Nevertheless, when the decision-making environment is fairly stable, hierarchical systems work adequately.

A service system that addresses multiple service processes and multiple value-creation opportunities necessarily introduces a need for coordinating multiple decisions by numerous DMAs. Hierarchies can exist within service systems. For example, within the health-care systems, there are corporate planners for hospital groups, administrators for individual hospitals, departments within each hospital, nursing teams, and so on. The service system modeler definitely needs to understand the mechanisms for hierarchical decision making. However, as service systems pursue cocreation of value in ever-broadening networks of actors and agents, hierarchies are not always feasible or advisable for decision making. Modern IT and social or professional networking provides a tremendous expansion of the opportunities for asynchronous and distributed decision making. Hence, in addition to hierarchies, service systems inherently involve *networked decision making*. In order to understand, model, and design the decision-making subsystems of a service system, we need to model decisions in detail, which is the subject of the next chapter.

Summary

- A decision is based on a cause–effect relation from alternatives and parameters to performance measures.
- Fast thinking is derived from slow thinking.
- Slow thinking is based on model building.
- Decision criteria reflect the structure of the actor's sense of value.
- A decision that involves a lot of alternatives, parameters, and performance measures is complicated but not complex.
- A decision with indeterminacy in the form of uncertainty in measuring parameters and predicting performance measures is complicated but not complex.
- A decision with indeterminacy in the form of vagueness or ambiguity in specifying alternatives, parameters, performance measures, or criteria is complex.
- Analytical techniques are available for modeling complicated and complex decisions.
- A typical service journey passes through abductive or dissonant, inductive or consonant, and deductive or resonant phases.
- Decision making in service systems is often coordinated through networked relationships instead of hierarchical relationships.

CHAPTER 7

Decision Analysis

Perspective

The previous chapters have established several key principles of the role of decision making in service systems:

- The service journey involves a sequence of decisions.
- Value propositions are cocreated throughout the service journey.
- Service termination, deviation or re-engagement are all possible within the service journey.
- Every service activity within a service journey potentially requires engagement decisions by all participating agents.

The ISPAR model of service (Maglio et al. 2009) identifies the essential junctures of a service journey as the steps of Interact- Serve-Propose-Agree-Realize. ISPAR takes a high-level view of a service, as opposed to the activity-level view required for service system modeling at an operational level. As the ISPAR model reflects the challenge of defining service-level agreements (SLAs), its structure is somewhat attuned to the *blueprint* approach to value propositions (see section "Rethinking: Value Propositions" in Chapter 3). For exchange valuation, this view of a service engagement as a "contract" (official or unofficial, written or verbal, explicit or implicit) is appropriate. An inspiration for this book, the ISPAR model is herein interpreted in the context of decision making and applied at a more granular level in order to capture the scope of decision mediation in a service journey. The realization that *actual* interacting, serving, proposing, agreeing, and realizing takes place with each service activity within a service journey compels a slightly different view. Therefore, we drive the ISPAR model to the tactical and operational level by proposing that the steps of ISPAR occur iteratively with each service activity and its engagement decision within a service journey as opposed to the service journey consisting of the five ISPAR steps.

For example, consider a company that must decide what specifications it should set for a SLA for support of its customer-relationship management (CRM) function. The SLA must delineate the deliverables of the vendor and the roles of company personnel in the provision of the consulting support for the company's CRM system, big data analytics, system expansion, and system integration with supply chain functions. The crafting of such an SLA is likely to be achieved through a sequence of discussions with the vendor. Through these discussions, both parties will learn about the nature of the context of the proposed service and adapt the commitments of the SLA accordingly. Furthermore, the de facto SLA is likely to evolve and change as the service is delivered. Hence, the construction of the SLA is itself a service journey, mediated by decisions. Each of these decisions is complicated, as the SLA requires the coordination of numerous capacitated resources and, in the case of a large company, an enormous number of options for service deliverables. Each of these decision may be complex, as the key performance indicators (KPIs) reflect uncertainties, vagueness, and ambiguities about the client firm's context and requirements and the vendor firm's capabilities. Other examples that clearly reveal the iterative decision guidance of the service journey are

- The progress of a consulting project through its myriad tasks and deliverables;
- A restaurant meal served through multiple courses;
- A web site that offers a user a menu of information resources and forms through links; and
- A medical treatment that proceeds through the phases of diagnosis, second opinions, surgery, medication, and therapy.

For simplicity of exposition, we will continue with an example of a student at a university, who may enroll in a course and pay tuition. The student's engagement in this educational service takes place through a long sequence of activities—classes, assignments, exams, projects, instructor meetings, and so on. Each step of the journey includes a service activity which requires a commitment of resources by each participating agent—decisions which are not necessarily coordinated or consistent. Each of these activities is preceded by engagement decisions by the student and the instructor, and the commitment of resources to the activity is the outcome of each agent's engagement decision. Throughout the service journey, an agent's level of commitment can vary or terminate entirely (e.g., the student stops attending classes). Whether or not the agent participates in an activity and the amount of committed resources are choices made by the agent. Of course, the success of each service activity is determined by these commitment decisions and, hence, the service journey is governed by a journey of decision making. Therefore, to reiterate a key point made in previous chapters, ongoing decision making by all participating agents in a service system direct the service journey.

We are forced to conclude that a service system modeler must be a decision modeler. Decision modeling is a discipline in its own right, which has reached a very advanced state of sophistication over the past 50 years. This chapter describes the structure of decision models and how decision models can capture even the most complicated and complex features of a decision. The reader is urged to study this chapter carefully with the aim of achieving an understanding of decision modeling sufficient to enable construction of decision models with the assistance of an analyst. Think of building the decision models for a service system as a service itself. In fact, decision analysts provide this very service, and if they do it right, they insist on the service system modeler taking ownership of the model building process. Therefore, every service innovator is required to be a decision modeler.

What Is a Decision Model?

As decisions are critical activities in the journey of a service, one cannot adequately model a service system without modeling the decisions that agents make within that system. A decision model is an analytical device such as a software application, spreadsheet, structured worksheet, mental paradigm, mathematical formula, or any rigorous specification of the cause–effect relationship that lies at the core of a decision. Every decision can be modeled. In this chapter, we present a brief overview of decision models. The reader who is interested in delving into the inner workings of a service system is strongly encouraged to study this chapter in detail. In the previous chapter, we examined the process of a decision-making agent (DMA). In this chapter, we examine the process of modeling how the agent makes decisions.

A surprisingly high percentage of people do not know how to define the structure of a decision. The inability to construct an analytical framework for a decision hampers both a decision maker and a decision modeler. Equally surprising is the fact that there is a general, mathematical structure to decision models which can be learned by anyone who is willing to consider an analytical approach to decisions. For the service system modeler, the application of this structure to the system model is an essential skill because, as we discovered in previous chapters, decision making is the driving force of service system journeys. We must admit that expressing a decision model in precise mathematical terms, invoking the appropriate optimization routine for finding an optimal solution and performing valid statistical analysis for the estimation of parameters does require education in some advanced mathematical subjects. However, the failure to implement decision models in practice has often been due to the gap in perspective between the DMA who understands the nature of the decision context and an analyst who understands decision modeling. Therefore, we urge all service system modelers, regardless of their level of mathematical training, to exercise patience in reading this chapter in order to become conversant in the language of decision modeling. If this degree of understanding can be achieved, then the service modeler can engage the support of an analyst in a fruitful cocreative model-building endeavor.

Smart Service is the new big thing! If you are reading this book, then it is likely that you are interested in participating in the emerging economy of smart service systems. What makes a service system smart? A lively debate about this question seems to emerge frequently even though the general answer is quite simple. A smart service system is one that incorporates automated decision support through the use of decision models. Later in this chapter, we identify different types of smartness based on the types of models that are used. For now, the reader only needs to realize that decision modeling is essential to the service systems that will bring about the next industrial revolution and that a conversational knowledge of the language of this discipline is a skill that the service system modeler cannot live without.

Making a decision always requires "constructing" a model. In most cases, this model construction is ad hoc, informal, and even subconscious to the decision maker. For example, the engagement decisions of common service encounters such as deciding to pursue the contents of a website or quit the site in favor of another is typically made quickly and with only a modest amount of conscious thought. For many familiar decision contexts, the decision maker executes a simple decision rule in order to select a course of action. However, even in these cases of routine decision making, the decision rules or heuristics that seem to automate the decision making are derived and adapted through the construction of a decision model. The service system modeler is tasked with understanding the engagement decisions that will determine the service journey of a service participant to the extent that the service system can be designed to support these decisions in the most rational manner.

All rational decisions have a structure and a process for arriving at a selected course of action. As service modelers, we must understand how a DMA can make a rational decision, even when the context involves uncertainty and risk or imprecise understanding of the problem. By modeling the rational choices that agents can make, we can better understand irrational choices and bounded rationality.

Agents must make decisions, even when the outcomes of each option cannot be predicted with certainty and when the agent's understanding of a problem is vague or ambiguous. Many people think that precise decision modeling cannot be applied to such cases, but the history of decision modeling (now more than 50 years) has produced numerous methods for representing these realities of decision making. First, there is the issue of uncertainty. Uncertainty in the parameters of decision leads to risk in the performance measures of the decision. Risk can be modeled, quantified, and traded off against other performance measures. In fact, this kind of trade-off is commonly made in both ad hoc decision making and analytical decision making. A model that explicitly represents uncertainty and risk is called stochastic. Stochastic models have been constructed and widely used in finance, marketing, operations, engineering, public policy, and every other field in which plans have to be made and systems have to be designed.

Another form of indeterminacy in decision making is the lack of understanding by the decision maker of the nature of the decision. Vagueness or ambiguity in the specification of decision options, parameters, or performance measures precludes a precise specification of a decision. However, even in cases in which the vagueness cannot be resolved, it is possible to build and apply a model of the decision using modeling mathematics known as “fuzzy.” Like stochastic models, fuzzy decision models have been used in all fields in which plans have to be made and systems have to be designed. The point is that decision modeling is applicable and beneficial in all walks of life and service systems are no exception.

Components of a Descriptive Decision Model

We begin by reviewing the three basic components of a decision model—decision alternatives, decision performance measures, and decision parameters. Recall from the discussion in the previous chapter that a decision is fundamentally about a cause–effect relationship. The causes fall into two categories: the controllable causes, which comprise the alternatives available to the DMA, and the uncontrollable causes, which are represented by parameters of the decision, which must be measured, estimated, or forecasted. The effects are the performance measures, which literally evaluate the performance of a chosen alternative.

We will use a simple example of a decision to illustrate the modeling concepts. Consider the decision of a student who must decide which courses to select for full-time enrollment at a university for a semester. The decision alternatives can be identified as the set of all combinations of courses that can be selected from the university’s catalog—an overwhelming number of possibilities. In considering any one of these combinations, the student evaluates the potential outcomes in terms of several performance measures: anticipated workload of the course selection, compatibility of the course selection with the student’s weekly schedule of other activities, progress toward the student’s chosen degree, satisfaction of prerequisite requirements of the university, satisfaction of the student’s personal interest in subjects, level of difficulty of the selection, and

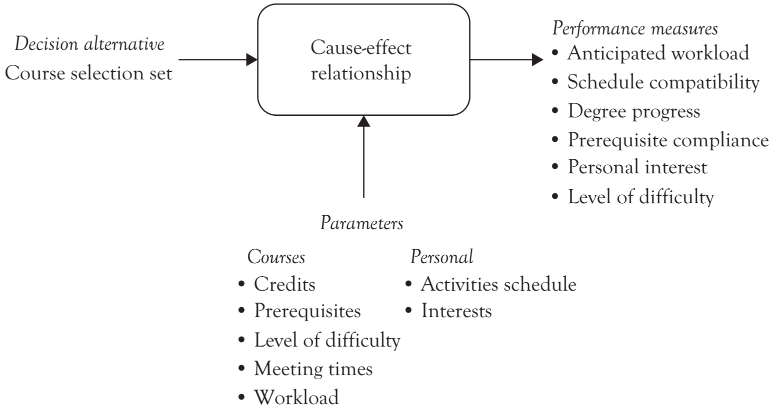


Figure 7.1 Example of the core decision relations

perhaps several other outcome measures of personal concern. Obviously, the student cannot make this evaluation without involving numerous data that influence the performance measures for a given selection, such as the number of credits allocated to each course, the prerequisites of each course, the relevance (required course, elective course, unrelated course) of each course to the student's degree program, the reputation of each course for its workload, the meeting times of each course, and so on. Figure 7.1 illustrates the three fundamental components of this decision and the relations among them, which we will define the *Core Decision Relations*. With this example in mind, we can proceed to formalize the specification of decision models.

Decision Variables

The first step in modeling a decision is that of specifying the options, alternatives, or choices that define the scope of a decision. Formally, we make this specification in terms of measurable quantities called decision variables. To be more specific, the modeler must be sure to define each decision variable with a well-defined name, units of measure, data type (binary, integer, real, etc.), and any restrictions on permissible values. A model of the enrollment decision shown in Figure 7.1 requires a precise data definition of any alternative course selection that might be considered. These variables would be specified in terms of binary decision variables to indicate which courses have been selected.

$$x_i = \begin{cases} 1, & \textit{if course i is selected} \\ 0, & \textit{otherwise} \end{cases}$$

Decision variables can be discrete or continuous. For example, a financial investment decision might be specified in terms of binary decision variables to identify the particular investment opportunities that are chosen as well as a set of continuous decision variables to represent the amount of money that will be invested in each opportunity. The most important feature of the specification of the decision variables is that these variables are capable of fully representing the scope of the decision, that is, the allowed values of the decision variables can identify every possible decision alternative.

Decision scope can be static or dynamic. Many plans and schedules are executed through a sequence of decisions over some time horizon. For example, the university student must make course selections for the current semester as well as course selections for each of the remaining semesters in the student's degree program. If the decision maker has the opportunity to discover some or all of the outcomes of previous decision in the sequence and adapt future decisions to these outcomes, the decision is labeled dynamic as opposed to static. Models of dynamic decisions begin with a recognition of a sequential decision process and the indexing of decision variables according to that sequence. In the case of the university student, a full specification of the decision variables for the entire degree program would be as follows:

$$x_{it} = \begin{cases} 1, & \textit{if course i is selected in semester t} \\ 0, & \textit{otherwise} \end{cases}$$

where $t = 1,2,\ldots,T$ and T = total number of semesters required for the degree program

Performance Measures

The second step in decision modeling is specifying the performance measures. Performance measures are the dimensions that evaluate the feasibility and desirability of any decision alternative. Similar to decision variables and parameters, performance measures must be specified with precise data definition statements. Each constraint and each objective of a decision is represented by a performance measure. For the construction of a model, the SMART principle should guide the specification of performance measures by which each performance measure should be defined with the following characteristics.

- **S**pecific—a well-defined name or label.
- **M**easurable—a unit of measure.
- **A**ttainable—realistic in its relation to the decision.
- **R**elevant—a function of the decision variables.
- **T**imely—a specification of the time of the attainment of the performance.

Parameters

The third step in decision modeling is specifying the parameters. Once the decision variables and performance measures have been specified, the decision modeler is faced with the task of expressing the relations that establish each performance measure as a function of one or more of the decision variables. However, this exercise of relation building inevitably leads the modeler to realize that the performance measures cannot be determined by the values of the decision variables alone. In addition to the decision variables that are under the control of the decision maker, the full determination of the performance measures requires other factors not under the control of the decision maker, which are represented by the parameters of the decision model. Only when the decision variables and performance measures are defined, can the modeler discover these missing factors.

Parameters are the influences on the performance measures that are not controlled by the decision maker, and they must be included in the descriptive model in order to complete the determination of the performance measures. As uncontrollable factors, parameters must be measured, estimated, or forecasted. As the example of the student's course selection decision illustrates, the parameters of a decision model represent relevant data extracted, transformed, and loaded (ETL) from external sources such as the university catalog, online recommendation systems, and word of mouth. As with decision variables and performance measures, the modeler must specify all parameters with precise data definitions. Only with this precision, can the data sources be identified and the ETL performed.

Key Performance Indicators

Time Summary

Some parameters, such as demand for a service, are functions of time. In other words, there is not a single parameter, but a time series of parameter values that represent the parameter. In some cases, the variation of a parameter over time follows a stable pattern. Such parameters are stationary. In other cases, this variation shows nonstationary patterns such as trend or cyclical behavior. The decision modeler must consider the effect of time-dependent behavior of parameters on the performance measures. Time-varying parameters imply time-varying performance measures, and, as a general rule, nonstationarity in parameters implies nonstationarity in performance measures.

Very often the effects of a decision are experienced long after the decision is made. For example, once the student selects the courses to be taken in a semester, the student is committed. Over the course of the semester, the workload, level of difficulty, and personal interest in the courses will rise and fall through the natural process of the course curricula. Then the decision maker and the decision modeler must determine how the entire time series of performance measure outcomes should be summarized into a single measure of the outcome of the decision. The decision maker has several approaches to summarizing the performance measures over time:

- Average
- Maximum
- Minimum
- Range
- Terminal value

Uncertainty

Uncertainty is an unavoidable fact of life in decision making and decision modeling. The unpredictability of life is reflected in parameter values that are not known with certainty. In this case, not only are parameters beyond the control of the decision maker, but the decision maker can only estimate or forecast the parameter values, with some attendant uncertainty.

In our example, the workload, the level of difficulty, and the personal interest satisfaction of each course cannot be known in advance. At best, the decision maker may be able to estimate a probability distribution for these parameters in order to measure the amount of uncertainty in their values.

The explicit representation of this uncertainty in the form of probability distributions classifies the decision model as a stochastic model. A stochastic decision model then maps the amount of uncertainty in parameters into the amount of unpredictability in the performance measures, the latter representing risk for the decision maker. Hence, the probability distributions of the parameters are translated into probability distributions of performance measures. In our example, the probability distribution of the workload of each course is translated by the decision model into a probability distributions of the total workload of the course selections. Similarly, the distributions for level of difficulty and interest satisfaction for each course are translated into probability distributions for the entire course set for the semester.

Many people express a bias about stochastic models. As soon as the prospect of not knowing each parameter of a decision with certainty arises, decision makers often wonder if there is any model-based approach to analyzing and advising the decision. The correct answer is, of course there is! In fact, model-based decision support systems abound in everyday life supporting risk-laden decisions in finance, marketing, supply chain management, energy grid management, government regulation, … the list is endless. How do these stochastic models provide solutions to risky decisions? The answer is no mystery. Stochastic decision models simply quantify rigorously the same kind of risk analysis that any human being would perform in facing a decision under conditions of uncertainty.

Risk

The foregoing discussion of uncertainty in parameters of a decision portrayed the decision model as translating the uncertainty in parameters to unpredictability in performance measures. This unpredictability we shall label risk. The term risk is used in many ways in common speech, and for the sake of rigor in our representation of decision models, we will herein adopt a more precise definition. To wit, we define risk as the randomness in performance measures and uncertainty as the randomness in parameters. The cause–effect relationship between parameters and performance measures (see Figure 7.1) implies that uncertainty engenders risk.

Almost every decision involves some risk. Parameters of a decision are elements that are measured, estimated, or forecasted. As such, one or more parameters of a decision are usually not known with certainty and correspondingly, the outcomes of any given alternative to a decision cannot be predicted with certainty. However, to an approximation, many decisions can be represented by models that suppress the uncertainty and risk. These models are called deterministic approximations. Any choice of alternative that is prescribed through the use of a deterministic model is useful only to the extent that the risk inherent in the decision is negligible or can be mitigated through supplementary actions.

Risk can be measured in a variety of ways by summarizing the effects of the possible scenarios for uncertainty. Typical risk measures fall into two categories, which, in the financial world, are called value at risk (VAR) and conditional value at risk (CVAR). Variations on these two summary measures of risk can be found in every problem domain. To represent these two approaches to quantifying risk as simply as possible, consider the probability distribution of the workload of a selection of courses chosen by a student.

Define,

$\phi(w)$ = probability density function of the number of hours of work, w, that will be required.

L = a parameter chosen by the student that represents the upper limit on what the student considers to be a comfortable workload.

$\int_L^\infty \phi(w)dw$ = a measure of risk that expresses risk as the probability that the workload exceeds the comfortable limit. This specification of risk is congruent with VAR measures of risk.

A disadvantage of the VAR-type measures of risk is that, although the decision maker knows the probability that the outcome of a performance measure may exceed a given threshold, the risk measure does not inform the decision maker about the extent of the transgression. For this reason, the CVAR risk measures are used, which are analogous to the following variation on the measurement of the effects of undesirable outcomes. Define,

$\int_{L}^{\infty} (w - L)\phi(w)dw$ = a measure of risk that expresses risk as the conditional expectation of the amount of excess workload.

Note that risk measures summarize the outcome of a performance measure over the random scenarios that are engendered by the randomness in the causative parameters. With such scenario-summarizing measures, the outcomes of an alternative can be expressed in terms of numerical values that capture the risk of a performance measure.

We have extended the specification of performance measures in two ways: time summaries for time-dependent performance measures and risk measures for summarizing the scenarios of possible outcomes of a performance measure that is a random variable. We will distinguish the summaries of performance measures (time summaries, scenario summaries, or both) from the performance measures themselves by referring to the former as KPI. In our example, we can see the possibility for both kinds of summaries. First, the workload of the course selection could be summarized over time by evaluating the average weekly workload in hours over the entire semester. This quantity is a random variable due to the uncertainty in the workload of each course. Therefore, the anticipated average weekly workload is a random variable for which the decision maker could choose either the VAR or CVAR type of risk measure. In this way, both kinds of summaries can be incorporated in a KPI.

At this point, our decision model must be called a "descriptive model" because it only describes the outcomes of any chosen alternative. This type of decision model, when codified in a decision support system provides the decision maker with a trial-and-error capability. That is, the model enables the decision maker to test any proposed alternative with respect to all relevant KPIs. Figure 7.2 illustrates the use of a descriptive decision model. Note that the user interface with the model has three components. The user inputs the values for the decision variables and parameters, and the model outputs the values of the KPIs.

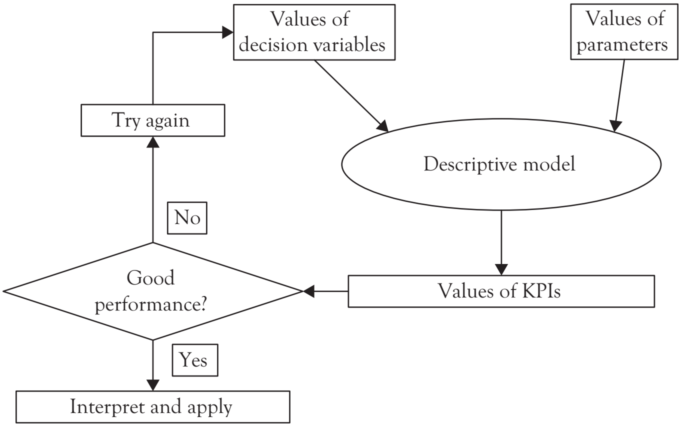


Figure 7.2 Descriptive model use

Components of Prescriptive Decision Model

For many decisions, such as the selection of a course set, the number of possible alternatives can be far too numerous to consider a manual trial-and-error approach to finding an optimal, feasible solution. Fortunately, operations researchers have been hard at work for many decades, developing and improving a wide variety of optimization methods that, in effect, take the burden of trial and error off of the decision maker. When a descriptive model is combined with an automated search algorithm that produces a recommended or prescribed solution, the model is called a prescriptive model.

Criteria

The descriptive model of a decision is the engine that drives the prescriptive model. Once the KPIs of a decision model have been specified and formulated, the decision maker must set the conditions that define an optimal, feasible decision. We call these conditions the decision criteria, which are expressed in terms of the constraints and objectives of the decision. KPIs should be defined such that each constraint is represented by a KPI and each objective is represented by a KPI. For example, the student who must select a course set for a semester might set the decision criteria as follows:

Maximize Expected Personal Interest
Subject to:
Workload Risk (VAR) ≤ 10 percent
Expected Average Weekly Free Time ≥ 10 hours
Degree Progress ≥ five courses
Prerequisite Requirements = 100 percent, for each course in the selection
Expected Total Level of Difficulty ≤ 7 (10-point scale)

In the final analysis, a decision-making actor (not the agent) evaluates the KPIs for an alternative against the actor's sense of overall value. The required or desired achievement levels of all of the KPIs embody the decision maker's trade-offs among the KPIs. Therefore, the decision criteria are, in effect, the decision maker's definition of value. Accordingly, the decision criteria are subjective and context specific.

Optimality Conditions

Ideally, the selection of a decision alternative is made by searching over all feasible solutions and finding the best one. This search for an optimal, feasible solution is usually guided by some mathematical conditions on this solution which serve to narrow down the search. These optimality conditions are derived through advanced mathematical operations and theory such as the Karush–Kuhn–Tucker (KKT) conditions for general cases as well as through model-specific analysis of the mathematical form of the KPIs as functions of decision variables and parameters. An exposé of these derivations is far beyond the scope of this book, as it comprises the subject of operations research and its manifold applications (see Hillier and Lieberman 2015).

Search Algorithms

There are many computerized algorithms for finding an optimal, feasible solution to decision problems. Perhaps the most widely known of these algorithms is the Simplex method of linear programming, which is taught to most college students in the fields of business or engineering and soon forgotten. In general, the word *programming* is used in operations research to indicate a search algorithm. Identifying a procedure for finding an optimal, feasible solution to a prescriptive model, the word *programming* was coined in the 1950s, and in the context of decision modeling, this word does not connote computer programming. As the search algorithms for finding these solutions must be customized for the mathematical structure of the descriptive model, there are numerous kinds of programming: Linear Programming, Integer Programming, Nonlinear Programming, Dynamic Programming, Geometric Programming, Constraint Programming, and so on. Most of these algorithms can be found as applications within user-friendly modeling software packages such as AIMMS (www.aimms.com), relieving the modeler of the optimization task. With such IT support, the modeler can focus on formulating the descriptive model and the criteria.

Decision Rules

Sometimes the optimality conditions can be expressed in the form of relatively simple set of formulas or computational procedures that prescribe a solution for a given set of parameter values. Such formulas can be called *decision rules*. When they exist, decision rules allow a shortcut to finding a solution to a decision. The DMA only has to enter the context parameters into the decision rule, and the prescribed alternative is selected. There is no need to perform an algorithmic search for a solution which would entail computing the KPIs for many different trial alternatives. Therefore, decision rules or even approximate decision rules offer dramatically more efficient methods for prescribing an alternative. In fact, once a decision rule is learned, the descriptive model no longer needs to be invoked.

Heuristics

A heuristic is a decision rule that may not guarantee an optimal solution in every context but can ensure the user that the solution will be acceptably close to optimum. In exchange for the potential deviation from optimality, the heuristic user gets a very efficient routine for finding an

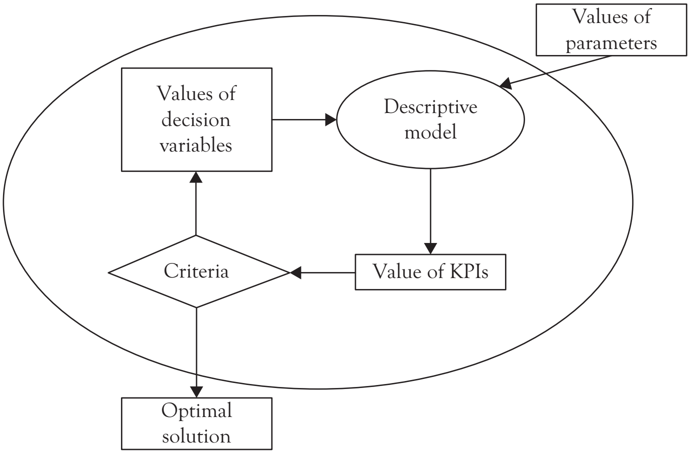


Figure 7.3 Prescriptive model use

answer to a decision and something else. In order to qualify as a heuristic, the decision rule must also be intuitive. Not all decision rules are heuristics. In other words, the DMA that executes the heuristic enjoys the confidence of using a method for finding a solution that makes sense.

Figure 7.3 illustrates the use of a prescriptive decision model. Note that the user interface with the model has only two components. The user inputs the values for the parameters, and the model outputs the prescribed solution.

The Core Decision Relations

Figure 7.4 shows the associations of the various elements of descriptive and prescriptive decision models. The reader should note that a prescriptive model cannot exist without a descriptive model. After all, the prescribed solution that is delivered by prescriptive model is the result of optimizing a descriptive model. Without a descriptive model, there is nothing to optimize. For this reason, the descriptive model is embedded within the prescriptive model. Also note that the fundamental cause–effect relationship from decision variables and parameters to performance measures is at the core of the descriptive model. Therefore, decision modeling begins with establishing these *Core Decision Relations*.

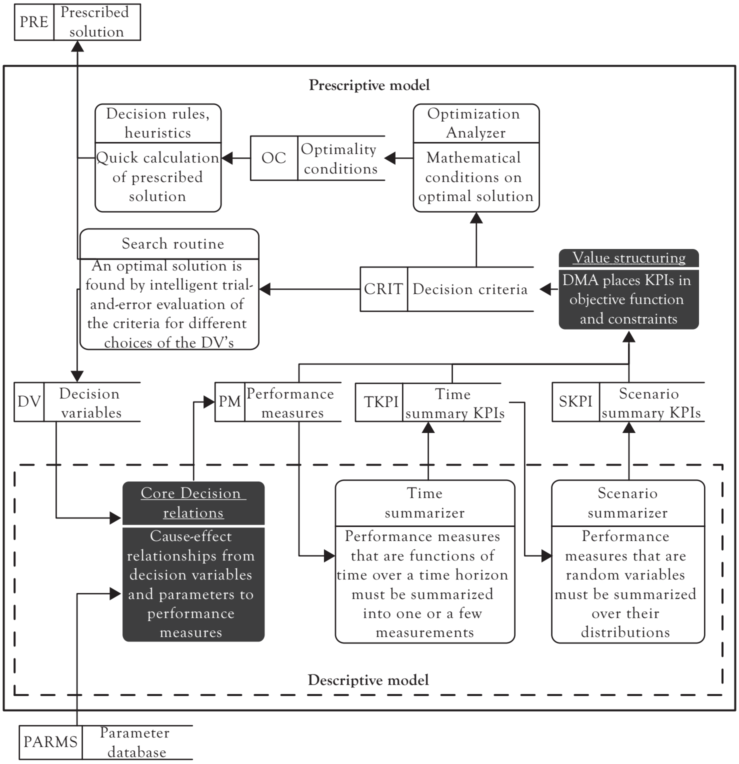


Figure 7.4 Descriptive and prescriptive model structure

Making this basic point is important because the industry that has built up around decision support systems has overshadowed this core requirement of such systems. The emphasis and the profits from software and hardware have always been on databases, data storage and retrieval, data manipulation, search algorithms, user interfaces, and so on. However, all of the information technologies that are necessary or useful in a decision support system are unable to fully inform a DMA without the Core Decision Relations being specified. All of the data that are collected, stored, extracted, transformed, and loaded exist for the purpose of building the parameter database that is required by the Core Decision Relations. All of the optimality conditions, search algorithms, decision rules, and heuristics that produce a prescribed solution require the Core Decision Relations and the Value Structuring as drivers.

The Human Element

Although generic performance-measure summaries, search algorithms, optimization principles, and heuristics are available, the Core Decision Relations are formulated specifically for the decision at hand and require the knowledge of a subject-matter expert (SME). Also, the Value Structuring, which specifies the decision criteria, requires the valuation of the KPIs by the Actor for whom the decision is made. These two components of the prescriptive and descriptive models are highlighted in Figure 7.4. Automation of decision support has made great advances in the past few decades, and the next decade promises to bring automation to almost every aspect of decision modeling. It is the Core Decision Relations and the Value Structuring that still requires the expertise of the DMA and the subjectivity of the Actor. Until computerized systems become smart enough to have the knowledge of an SME and the ability to know an Actor's motive at least as well as the Actor, these two necessary activities of decision analysis will have to be performed by humans.

Predictive Decision Models

Predictive models turn structured data into information. The word *predictive* is something of a misnomer because the function of predictive model in supporting decisions is to estimate the parameters of the Core Decision Relations. Prediction or forecasting is only one form of such estimation, so the term predictive model connotes an inaccurately narrow representation of this type of model.

The reader should note that a predictive model has no purpose without a descriptive model. After all, the defining characteristic of information is its usefulness to a decision maker. Information is not noise. Information that supports a decision is defined precisely by the parameters of the Core Decision Model, as these parameters express knowledge of the cause–effect relations that are of interest. Without a descriptive model, there is no way to identify information. For this reason, the predictive model is built to support the Core Decision Relations.

Data

Many, if not most, analysts begin model building with data. This approach can be a big mistake. Somewhat surreptitiously, the data can become the modeler's guide instead of the problem that needs to be solved, a phenomenon well known in conventional statistical research and now emerging as a trap of big data analytics. Model discovery through random data searching can easily consume intolerable amounts of time and effort. Therefore, the modeler is advised to drive the quest for data by the needs of the decision model that the data supports. In other words, the Core Decision Relations should be built in order to discover the parameters that need to be measured, estimated, or forecasted, which in turn will point the way to the appropriate data.

The burgeoning use of Big Data has only amplified the distorted role of data and data analysis in decision modeling. Big Data opens up vast stores of data that previously were unavailable to or unheard of by modelers. Besides offering access to large samples of data, the Big Data revolution provides an ever widening array of variables that can be measured. However, the data is useful to the decision modeler only to the extent that the data can measure, estimate, or forecast the parameters of a decision model. Therefore, in order to put data analytics in its proper place, the modeler needs to guide any empirical efforts with a hypothesized descriptive model or at least a framework for such a model.

What then is the role of data in building decision models? There are three stages in the use of predictive modeling: descriptive model specification, descriptive model validation, and descriptive model estimation. These three stages correspond to the abductive, inductive, and deductive stages of decision making that were described in Chapter 6. Data analytics supports all three of these stages in different ways. However, the process of model building is owned by the DMA and data analytics must be put to work on behalf of the DMA (instead of the other way around as often happens in practice).

Model Specification

The word "specification" is used to identify the complete set of definitions of decision variables, performance measures, KPIs, and parameters of a decision model as well as the Core Decision Relations that link the causative factors of decision variables and parameters to the performance measures. At the outset of building a decision model, either formally or informally, the DMA may not be aware of these basic elements of the decision. The scope of the decision in terms of kinds of alternatives that are available may not be clear. For example, an entering freshman student may not be aware of the structure of the semester program that must be chosen or the way that this program is expressed. The student may not fully grasp all of the performance measures that are relevant to the program choice and requires some introspection and learning about the nature of the university catalog, prerequisite rules, and so on. In other words, the form of the decision model is not fully known to the DMA. Empirical study can illuminate the dimensions of the decision, but at this abductive stage of decision modeling, the data analytics support does not necessarily have to involve extensive computation or statistical estimation. The DMA is only trying to determine the precise definitions of decision variables, performance measures, and parameters that are relevant to the decision. Exposure to the domain of practice that is relevant to the decision, question and answer sessions with domain area experts, literature review, and fuzzy reasoning are all analytical tools at this stage of model development.

Unstructured or unsupervised learning is essentially a way to discover the unknown causes in a cause–effect relationship. In the model specification stage of model development, unsupervised machine-learning algorithms can determine relationships among proposed or fuzzily defined decision variables, performance measures, and parameters. Although the accomplishments of machine learning are impressive, the model builder should be careful not to invest in such technology when simple human intuition and experience can identify cause–effect relationships. For example, the student who must select a course schedule should know that the formula for the performance measure of the total workload should be simply the sum of the workloads of the individual course in the schedule. On the other hand, gauging the interest level of a schedule could require a nonmetric examination of the student's background, expressed preferences for reading material, backgrounds of course instructors, and other factors. The model for measuring level of interest could be quite complicated and complex and require the Actor's participation in the modeling effort to express subjective trade-offs. In this case, machine learning could be very helpful in suggesting numerous formulations of this part of the decision model. The end result of the model specification phase of modeling is a set of data definitions for the decision variables, performance measures, and parameters of the model and the Core Decision Relations.

Model Validation

Model validation determines if there is substantial evidence to believe the cause–effect relationship that a descriptive model specifies. Performing this validation requires collection of a sample of data of all of the decision variables, performance measures, and parameters and performing statistical tests to determine whether or not these elements are related in the manner that was hypothesized by the model specification. In validation, the decision modeler performs the inductive reasoning of hypothesis testing following the well-established path of the scientific method. Here predictive modeling has an important and traditional role to play. Recall that the parameters of the model must be measured, estimated, or forecasted, and these calculation are by-products of the validation analytics.

Data analytics can be seductive. In many decision-modeling cases, the decision environment is novel or has some unique features. This is especially true in service innovation, when engagement decisions are individualized and context dependent. In this case, statistical validation is not possible, at least for some cause–effect relations within the model and the modeler is left to rely on model verification—the inspection of the logic by which these relations are hypothesized. Also, when data is lacking, the best that a modeler can do is use judgment to estimate parameters. Although every modeler is uncomfortable without strong data support for a model, the well-known adage of operations researchers should be kept in mind: "It is better to solve the right problem approximately than to solve the wrong problem to a high degree of precision and accuracy."

Data Models and Devices

In the previous section, we applied a traditional interpretation of the term *predictive model* to state that a predictive model converts structured data into information. In recent years, the burgeoning technologies of Big Data Analytics has brought unstructured data analysis into the role of producing information in the form of useful parameter estimates in support of decision models. Accordingly, many analysts apply the term *predictive modeling* to encompass unstructured data analysis. However, the conversion of unstructured data such as text and images into parameter estimates requires an intermediate step of converting unstructured data into structured data. The science and technologies for this kind of conversion are recently developed and have nothing in common with the traditional statistical models used in the predictive modeling that is discussed in the previous section. Therefore, we opt to classify the data conversion models in a category of their own which, in the interest of clarity, we shall call Data Models. Much of the effort in the field of Big Data Analytics can be assigned to this class.

The reader should note that a Data Model has no purpose without a Predictive Model. After all, the reason for structuring data is to support its use in generating information. Consequently, the Data Model is built to support the Predictive Model.

Devices have also established a new component of decision support systems. Smart devices such as infrared sensors, high-resolution video cameras, biometric monitoring devices, wearable chips, environmental monitors, eye-scan monitors, WAAS-enabled GPS receivers, and many others are now generating an avalanche of Big Data that form the raw material of smart service systems. Continuing the hierarchy of model components that we have built so far in this chapter, we can see smart devices given purpose by the need for massive amounts of unstructured data. In other words, the smart devices support the data models, which in turn support the predictive models, which in turn support the descriptive models, which in turn support the prescriptive models. Smart devices are also found at the end of the decision modeling hierarchy as prescribed solutions can be executed with automated systems for driving cars, flying

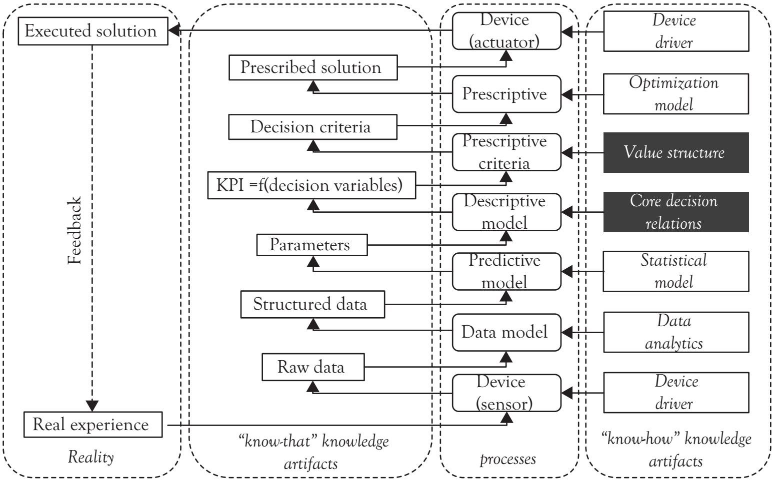


Figure 7.5 Model hierarchy

airplanes, moving prosthetic limbs, and many other robotic systems for actuating decisions. Figure 7.5 illustrates this model hierarchy.

Big Data and Machine Learning

Each stage of the decision that is shown in Figure 7.5 is an activity that transforms one knowledge artifact into another. Each of these artifacts represents a *know-that* knowledge resource. Hence, each stage of the decision itself can be modeled as a service activity (see Chapter 3). Like any activity, each of these activities requires some kind of code, algorithm, procedure, method, framework, or human judgment as an input. Figure 7.5 shows these *know-how* knowledge resource inputs in the right-most column of elements.

Know-How Versus Know-That Knowledge

Know-how knowledge has many forms. For example, an algorithm, such as the simplex method for linear programming, could be one of the supporting knowledge artifacts for the prescriptive modeling stage. Even better, a library of optimization routines and an intelligent server to assign the most appropriate algorithm to the decision model at hand comprises a modern optimization "engine" such as AMPL (ampl.com), GAMS (www.gams.com), CPLEX (http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html), and others. Software packages that support statistical analysis such as R (www.r-project.org), SPSS Modeler (http://www-01.ibm.com/software/analytics/spss/products/modeler/), SAS (www.sas.com), and others provide the know-how for predictive modeling. However, we should hasten to add that the application of these knowledge artifacts generally requires the addition of expert knowledge from a human analyst. Therefore, decision activities are among the most sophisticated activities within a service system.

Clearly, the know-how knowledge resources embody complicated, and perhaps, complex operant dimensions (see Chapter 6). For example, statistical analysis in support of predictive modeling can require large, complicated software packages. We recognized in Chapter 6 that vague understanding of the dimensions of value is the prime example of complexity in knowledge. In this particular case, this complexity affects the prescriptive and descriptive activities of decision making (see Figure 7.5). Data analytics in support of data models usually require some complex human judgment for validation. Therefore, know-how is derived from intelligence and learning.

The term *smart service system* is now bandied about the community of data scientists without much agreement about just what makes a system smart. The central concept behind smartness is the active role of intelligent machines in providing the know-how for decision activities. These roles can be many and varied, which leads to different kinds and degrees of smartness. Hence, smartness is not a one-dimensional measure. In the rest of this chapter, we shall establish a position in this debate with a specification of *smartness*.

Extant Knowledge Versus New Knowledge

How is the know-how acquired and retrieved for the support of a decision? Figure 7.5 identifies the type of process knowledge that is required for each decision-modeling stage. Most of the process knowledge in Figure 7.5 has been developed over many decades of research. Optimization engines, statistical methods, text analytics, and device drivers apply codified knowledge that is the result of decades, even centuries, of scientific research, engineering design, and technology development. These knowledge resources exist in more or less generic forms and can be applied automatically to any decision within an appropriately bounded domain of application. Such extant knowledge enables the automation that is commonplace in professional and personal decision support systems. These knowledge resources have certainly imbued smartness into the decision support systems and the larger service systems within which the decision making is embedded. However, these kinds of smartness have been deployed for several decades, long before the term smartness became trendy.

What then is the notion behind the current excitement over the creation of smart systems? The process knowledge required for formulating the descriptive model and the value model is context specific and must be identified and acquired by the service system. Now, we can see a different kind of know-how knowledge that is required for decision support. This kind of knowledge does not exist before the service system begins its work. New knowledge that is context specific must be created. The system, specifically the agents and actors within the system, must learn this knowledge. Traditionally, such learning could be done only with the intervention of a human modeler who facilitates the construction of the descriptive model and the value model for the actors involved in the service system. However, the current age of IT advancement now enables machine learning, and what the machines are directed to learn is the formulation of the descriptive decision model and the actors' value models. In the current debate over smartness, it is precisely this capability that earns a system the label *smart*. Even a modest discussion of machine learning and big data analytics is far beyond the scope of this book. For our purposes here, we only require an understanding of where in decision support these capabilities can be utilized.

Adaptation and Learning Systems

Learning and Model Adaptation

An old adage claims, "Good judgment comes from experience, and experience comes from bad judgment." DMAs, and particularly agents that engage a service system, are generally faced with a sequence of decisions. In the case of engagement decisions of a service, the agent can be afforded a large number of engagement decisions. For example, the student taking a course can decide at least once per day how much of a commitment he or she will make to the course in the next day and beyond. A customer in a restaurant can decide after each course whether or not to continue the meal or what to order for the next course. A consultant hopes to convince a client to continue the consulting service at each point in the contract at which the client can cancel the service. The provider of online IT support service to companies is aware of the client companies' right to cancel or expand the service at any time. In all such cases, the service participants have the opportunity to use experience with service engagements to inform the ensuing engagement decisions. Formally, this process consists of updating decision models based on experience and data collection. In other words, learning from the outcomes of previous decisions enables the DMA to adapt the decision models of future engagement decisions. Clearly, if a service journey is to be successful, every DMA's learning and adaptation must be considered part of the service system design.

We can formalize the learning process with specific and well-defined procedures for updating the decision models of a service journey. Feedback of information is the instigator of updated decision models. A DMA will update a decision model in response to two kinds of experience: data collection and performance of previously applied decision models. Returning to the example of the student in a semester-long course, we can see how progress through the course changes the state of the student and the upcoming workload. Naturally, any future engagement decisions should be based on the updated state of course progress. In addition to updating the state of the student, the parameters of the student's model for workload may require updating based on how accurately previous applications of decision models predicted performance measures. The student may have discovered that the parameter of hours of study per class period may have to be increased or decreased based on the experience to date. Once the process of adaptation is understood, the service innovator can design the sequence of service activities and the type of feedback provided after each activity in such a way that the learning and adaptation is supportive of value-generating journeys.

We can define three broad categories of learning and adaptation in the context of decision model building: abductive adaptation, inductive adaptation, and deductive adaptation. Note that these categories mirror the three stages of model building because the process of updating a model is essentially the same as the process of building a model from scratch. The difference between building and updating, however, lies in the extent of the analytical work that needs to be done and the order in which it is done. Ordinarily, the original construction of a model proceeds through the abductive, inductive, and deductive stages in that order. However, updating a model generally takes place in the reverse order because DMAs generally minimize the effort required for adaptation and the difficulty of the adaptation is lowest for deductive adaptation and highest for abductive adaptation.

We define and distinguish the three forms of adaptation as follows:

- Deductive adaptation simply updates the state of the decision maker and the decision environment based on the outcomes of previous decisions.
- Inductive adaptation updates the estimates of parameters of decision models based on the larger sample afforded by data collection.
- Abductive adaptation reformulates the specification of the decision model in consideration of data collection and the performance of the decision model.

Deductive Adaptation

In deductive adaptation, the agent is guided by a validated model of the cause–effect relationship of the next engagement decision. There are several forms of deductive adaptation, and we begin this discussion with the most basic form—open-loop or nonadaptive decision modeling. As the previous engagement in the journey is completed through which the state of the agent is changed, the DMA has the option of entering the next engagement without making any changes to the decision model for this engagement. In other words, no adaptation is performed. In the case of a well-understood decision that has a perfectly controllable outcome, this type of decision making makes sense. If there were no random influences on previous decisions and the parameters of ensuing decisions are unaffected by the passage of time or by the experience of previous decisions, then the nonadaptive approach makes sense. This type of decision making is often called "open loop" because there is no need for feedback information as shown in Figure 7.5.

The next level of deductive adaptation in sequential decision making responds to the deviations in the outcome of a previous decision due to random influences on parameters that cannot be controlled. This level of adaptation updates the status of the decision environment before updating the decision. The status of the decision environment is the result of the outcomes of the random parameters on the performance measures of the previous engagement. A service system that supports outcome-adaptive decision making must ensure that the DMA has feedback data that measure the deviations of predicted outcomes from actual outcomes.

In the example of the student engaging in courses during a semester of study, deductive adaptation would be executed in the form of updating the student's progress in each course. In making the next engagement decision, the student must evaluate the current state of his or her journey through the service system. If the student has fallen behind the planned accomplishments for some courses, then the state of the journey raises the challenge and workload requirements for the next engagement. Similarly, if the journey has advanced ahead of planned accomplishments, the state of the journey relieves some of the upcoming requirements.

Inductive Adaptation

In the inductive phase of decision making, the agent is guided by a valid model of the cause–effect relationship of the decision but with imperfect estimates of the parameters of this decision model. Inductive adaptation takes place when the parameters of the model of the Core Decision Relations are updated as new data become available. This level of adaptation utilizes a predictive model to update estimates of parameters with the result or reducing sampling error with every update. A service system that supports this estimation-adaptive decision making must ensure that the agent has all of the data and information that measure the parameters of the decision as well as a predictive model for converting the data into parameter estimates.

In the example of the student engaging in courses during a semester of study, inductive adaptation could be executed in the form of updating the student's estimates of the workload for each course. At the beginning of the semester, the student has no data on which to estimate the number of hours per week that each course will require. As the semester progresses, the student will accumulate an increasing sample of data for estimating this important parameter. If the updated estimates are used in remodeling the commitment decisions of the student, then the student is executing an inductive adaptation.

Abductive Adaptation

Abductive adaptation of a decision model is the process of reformulating the model of the Core Decision Relations. At this level of adaptation, the agent does not possess a valid model of the Core Decision Relations and even the definitions of the basic components of decision variables, performance measures, and parameters are not fully known. A service system that supports specification-adaptive decision making must ensure that the agent has a workspace to experiment and investigate the opportunities for action and the potential outcomes of those actions. The aim of abductive adaptation is the reduction in the fuzziness in the DMA's understanding of the decision problem at hand. In the example of the student engaging in courses during a semester of study, abductive adaptation could be executed in the form of considering a change of major or rethinking the student's motives for attending college.

Figure 7.6 illustrates the three stages of learning and adaptation in terms of decision model support.

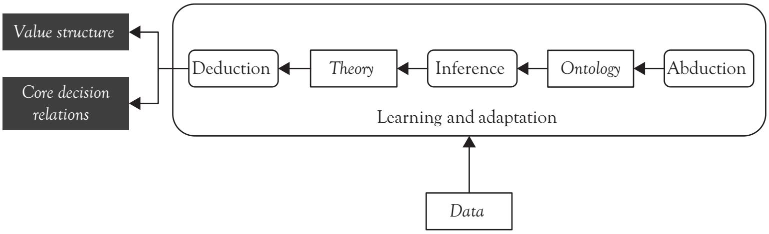


Figure 7.6 Adaptation for model building

Smart Systems

The most exciting development in the design of service systems is the intelligent decision support provided by cognitive systems. The term *Smart Service System* has become popularized to identify decision support systems that engage the so-called cognitive agents. However, the rapid development of these systems, most notably represented by the IBM machine known as Watson, has engendered some confusion about the precise capabilities of cognitive agents and smart systems. From the earlier discussion about the hierarchy of models, we can suggest a precise characterization of smart systems. We begin by realizing that there are different levels of smartness in systems so that it makes no sense to say that a system is smart or not smart. Instead we offer the following definition of smartness. A system is smart with regards to some decision function if that function is automated within the *system.* Then Figure 7.5 clearly indicates the different kinds of smartness that are possible. We can identify 18 categories or dimensions of smartness in accordance with the level of learning that the system undertakes and the decision model component that is adapted through this learning. Table 7.1 lists the resultant 18 dimension of smartness. Any system can be smart in any combination of these forms of smartness. Consistent with the difficulty and breadth of learning that is involved, we can say that inductive adaptation is smarter than deductive adaptation and abductive adaptation is smarter than inductive adaptation.

Table 7.1 Dimensions of smartness

		Level of adaptation		
		Deductive	Inductive	Abductive
Decision model component	Device model			
	Data model			
	Statistical model			
	Descriptive decision model			
	Value structure model			
	Optimization model			

Summary

- A decision model is an analytical device such as a software application, spreadsheet, structured worksheet, mental paradigm, mathematical formula, or any rigorous specification of the cause–effect relationship that lies at the core of a decision.
- Every decision can be modeled.
- Making a decision always requires "constructing" a model.
- The *Core Decision Relations* and *Value Structure* are the foundations of every decision model. All other model components are given purpose by these two foundations.
- A system can be smart in several ways by automating one or more components of the model hierarchy at different levels of learning and adaptation.

References

Allee, V. 2009. "Value Creating Networks: Organizational Issues and Challenges." *The Learning Organization Special Issue on Social Networks and Social Networking* 16, no. 6, pp. 427–42.

Alter, S. 2011. "Making a Science of Service Systems Practical: Seeking Usefulness and Understandability While Avoiding Unnecessary Assumptions and Restrictions." In *The Science of Service Systems*, eds. H. Demirkan, J.C. Spohrer, and V. Krishna, 61–72. New York: Springer.

Badinelli, R.D. 2010. "A Stochastic Model of Resource Allocation for Service Systems." *Service Science* 2, no. 1, pp. 68–83.

Badinelli, R. 2012. "Fuzzy Modeling of Service System Engagements." *Service Science* 4, no. 2, pp. 135–46.

Baines, T., and H. Lightfoot. 2013. *Made to Serve: How Manufacturers Can Compete Through Servitization and Product Service Systems.* West Sussex, UK: John Wiley & Sons.

Barile, S. 2009. *Management Sistemico Vitale*. Vol. 1. Torino: G. Giappichelli.

Barthold, D., A. Nandi, J. Rodriguez, and J. Heymann. 2004. "Analyzing Whether Countries Are Equally Efficient at Improving Longevity for Men and Women." *American Journal of Public Health* 88, no. 5, pp. 2163–69.

Bertalanffy, L. 1972. "The History and Status of General Systems Theory." *The Academy of Management Journal* 15, no. 4, pp. 407–26.

Bitner, M., B.H. Booms, and L.A. Mohr. 1994. "Critical Service Encounters: The Employee's Viewpoint." *Journal of Marketing* 58, no. 4, pp. 95–106.

Bitner, M., A. Ostrom, and F. Morgan. 2008. "Service Blueprints: A Practical Technique for Service Innovation." *California Management Review* 50, no. 3, pp. 66–94.

Bureau of Economic Analysis. 2015. "Gross-Domestic-Product-(GDP)-by-Industry Data." http://bea.gov/industry/gdpbyind_data.htm (accessed June 24, 2015).

Cardoso, J., and C. Pedrinaci. February 4–6, 2015. "Evolution and Overview of Linked USDL." In *6th International Conference Exploring Services Science*, eds. Novoa, Henriqueta, Dragoicea, and Monica, LNBIP, Vol. 201. IESS 2015, Porto, Portugal.

Chan, W., and C. Hsu. 2009. "Service Scaling on Hyper-Networks." *Service Science* 1, no. 1, pp. 17–21.

Chandler, J., and S. Vargo. 2011. "Contextualization and Value-in-Context: How Context Frames Exchange." *Marketing Theory* 11, no. 1, pp. 35–49.

Chesbrough, H., and J. Spohrer. 2006. "Research Manifesto for Service Science." *Communications of the ACM* 49, no. 7, pp. 35–40.

CIA. 2015. "GDP Composition by Sector of Origin, World Fact Book." https://www.cia.gov/library/publications/the-world-factbook/fields/2012.html (accessed June 24, 2015).

Collins, G., and D. Sisk. 2015. *API Economy, from Systems to Business Services*. Westlake, TX: Deloitte University Press.

Demirkan, H. 2015. "Enhancing E-Commerce Outcomes with IT Service Innovations." *International Journal of Electronic Commerce* 19, no. 3, pp. 2–6.

Demirkan, H., and D. Delen. 2013. "Leveraging the Capabilities of Service-Oriented Decision Support Systems: Putting Analytics and Big Data in Cloud." *Decision Support Systems* 55, no. 1, pp. 412–21. Demirkan, H., J. Spohrer, and V. Krishna, eds. 2011. *The Science of Service Systems*. New York: Springer.

Ferrario, R., N. Guarino, C. Janiesch, T. Kiemes, D. Oberle, and F. Probst. 2011. "Towards an Ontological Foundation of Service Science: The General Service Model." *10th International Conference on Wirtschaftsinformatik*. Zurich, Switzerland.

Finkelstein, S., R. Larson, K. Nigmatulina, and A. Teytelman. 2015. "Engineering Effective Responses to Influenza Outbreaks." *Service Science* 7, no. 2, pp. 119–31.

Garber, A., and J. Skinner. 2008. "Is American Health Care Uniquely Inefficient?" www.nber.org/papers/w14257 (accessed June 24, 2015).

Geerts, L., and W. McCarthy. 2002. "An Ontological Analysis of the Primitives of the Extended-REA Enterprise Information Architecture." *The International Journal of Accounting Information Systems* 3, no. 1, pp. 1–16.

Golinelli, G.M. 2010. *Viable Systems Approach (VSA) Governing Business Dynamics*. Milan: Wolters Kluwer Italia Srl.

Gordijn, J., and H. Akkermans. 2001. "Designing and Evaluating E-Business Models." *IEEE Intelligent Systems* 16, no. 4, pp.11–17.

Gronroos, C. 1994. "Quo Vadis, Marketing? Toward a Relationship Marketing Paradigm." *Journal of Marketing Management* 10, no. 5, pp. 347–60.

Hastings, H., and J. Saperstein. 2014. *Service Thinking: The Seven Principles to Discover Innovative Opportunities.* New York: Business Expert Press.

Hefley, B., and W. Murphy, eds. 2008. *Service Science, Management and Engineering*. New York: Springer.

Hillier, F., and G. Lieberman. 2015. *Introduction to Operations Research.* 10th ed. New York: McGraw-Hill.

Horkoff, J., A. Borgida, J. Mylopoulos, D. Barone, L. Jiang, E. Yu, and D. Amyo. September 10–14, 2012. “Making Data Meaningful: The Business Intelligence Model and Its Formal Semantics in Description Logics.” In *Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2012*, Rome, Italy. Proceedings, Part II, Berlin, Germany: Springer.

IBM Research. 2004. “Services Science: A New Academic Discipline?” https://www.ibm.com/developerworks/community/wikis/form/anonymous/api/wiki/ad5cdad0-33a1-46bc-8cba-2102de0279ff/page/4c6bbe0b-577d-4eef-af47-5d62a83a3ad4/attachment/bc1aa402-1369-46ef-a69b-bf725fe1a9f9/media/facsummit.pdf (accessed September 22, 2015).

Kahneman, D. 2011. *Thinking Fast and Slow*. New York: Farrar, Strauss & Giroux.

Kastalli, I., and B. Looy. 2013. “Servitization: Disentangling the Impact of Service Business Model Innovation on Manufacturing Firm Performance.” *Journal of Operations Management* 31, no. 4, pp. 169–80.

Karmarkar, U. 2004. “Will You Survive the Services Revolution?.” *Harvard Business Review* 82, no. 6, pp. 100–107.

Krigsman, M. 2009. “Study: 68 Percent of IT Projects Fail.” http://www.zdnet.com/article/study-68-percent-of-it-projects-fail/ (accessed June 24, 2015).

Lay, G., ed. 2014. *Servitization in Industry*. New York: Springer.

Lessard, L. 2015. “Modeling Value Cocreation Processes and Outcomes in Knowledge-Intensive Business Service Engagements.” *Service Science* 7, no. 3, pp. 181–95.

Maglio, P., ed. 2015. *Service Science* 7, no. 3, INFORMS.

Maglio, P., C. Kieliszewski, and J. Spohrer. 2010a. *Handbook of Service Science*. New York: Springer.

Maglio, P., S. Nusser, and K. Bishop. 2010b. “A Service Perspective on IBM’s Brand.” *Marketing Review St. Gallen* 27, no. 6, pp. 44–48.

Maglio, P.P., and J. Spohrer. 2008. “Fundamentals of Service Science.” *Journal of the Academy of Marketing Science* 36, no. 1, pp. 18–20.

Maglio P.P., S.L. Vargo, N. Caswell, and J. Spohrer. 2009. “The Service System Is the Basic Abstraction of Service Science.” *Information System and E-Business Management* 7, no. 4, pp. 395–406.

McCarthy, W. July 1982. “The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment.” *The Accounting Review* 57, no. 3, pp. 554–78.

Mele, C., and F. Polese. 2011. “Key Dimensions of Service Systems in Value-Creating Networks.” In *The Science of Service Systems*, eds. H. Demirkan, J.C. Spohrer, and V. Krishna, 37–59. New York: Springer.

Mora, M., M. Raisinghani, O. Gelman, and M.A. Sicilia. 2011. "Onto-Servsys: A Service System Ontology." In *The science of Service Systems*, eds. H. Demirkan, J.C. Spohrer, and V. Krishna, 151–174. New york: Springer.

NBER (National Bureau of Economic Research). 2015. "The Relative (In)-Efficiency of the U.S. Health Care System." www/nber.org/bah/2008no3/w14257.html (accessed June 24, 2015).

Ng, I. 2013. "Hat for Discovery of Contextual Archetypes." hubofallthings.com/h-a-t-for-discovery-of-contextual-archatypes/ (accessed June 24, 2015).

Ng, I. 2014. *Creating New Markets in the Digital Economy Value and Worth*. New York: Cambridge University Press.

Ng, I., R. Maull, and L. Smith. 2011. "Embedding the New Discipline of Service Science." In *The Science of Service Systems*, eds. H. Demirkan, J.C. Spohrer, and V. Krishna, 13–35. New York: Springer.

Ng, I., R. Badinelli, P. Dinauta, S. Halliday, H. Lobler, and F. Polese. 2012a. "S-D Logic: Research Directions and Opportunities: The perspective of Systems, Complexity and Engineering." *Marketing Theory* 12, no. 2, pp. 213–17.

Ng, I, L. Smith, and S. Vargo. 2012b. "An Integrative Framework of Value." www.academia.edu/1367426/Ng_ICL_Smith_LA_and_Vargo_SL_2012_An_Integrative_Framework_of_Value._WMG_Service_Systems_Research_Group_Working_Paper_Series_03_12 (accessed June 24, 2015).

Noran, O.S. 2004. "UML® vs IDEF: An Ontology-Oriented Comparative Study in View of Business Modelling." In *Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS 2004)*, eds. I. Secura et al., Vol. 3, pp. 674–82. Porto, Portugal.

OMG (Object Management Group). 2011. "Business Process Model and Notation." http://www.omg.org/spec/BPMN/2.0/PDF/ (accessed June 29, 2015).

OMG (Object Management Group). 2014a. "Introduction to OMG's Unified Modeling Language™ (UML®)." http://www.omg.org/gettingstarted/what_is_uml.htm (accessed September 23, 2015).

OMG (Object Management Group). 2014b. "Value Delivery Modeling Language." http://www.omg.org/spec/VDML/1.0 (accessed June 29, 2015).

OECD (Organisation for Economic Co-operation and Development). 2014. "Education at a Glance 2014." http://www.oecd.org/edu/Education-at-a-Glance-2014.pdf (accessed June 24, 2015).

Ostrom, A.L., M.J. Bitner, S.W. Brown, K.A. Burkhard, M. Goul, V. Smith-Daniels, H. Demirkan, and E. Rabinovich. 2010. "Moving Forward and Making a Difference: Research Priorities for the Science of Service." *Journal of Service Research* 13, no. 1, pp. 4–36.

Petrie, C., A. Hochstein, and M. Genesereth. 2011. "Semantics for Smart Services." In *The Science of Service Systems*, eds. H. Demirkan and J. Spohrer, 91–105. New York: Springer.

Qiu, R.G. 2009. "Computational Thinking of Service Systems: Dynamics and Adaptiveness Modeling." *Service Science* 1, no. 1, pp. 42–55.

Qiu, R. 2013. "We Must Re-Think Service Encounters." *Service Science* 5, no. 1, pp. 1–3.

Qiu, R. 2014. *Service Science: The Foundations of Service Engineering and Management*. New York: Wiley.

Regan, C. 1997. *A Concurrent Engineering Framework for Apparel Manufacture* [dissertation]. Virginia Polytechnic Institute and State University, ProQuest, UMI Dissertations Publishing. (UMI No. 3317771).

Rumbaugh, J., I. Jacobson, and G. Booch. 1999. *The Unified Modeling Language Reference Manual*. Reading, MA: Addison-Wesley.

Rust, R. 2004. "A Call for a Wider Range of Service Research." *Journal of Service Research* 6, no. 3, p. 211.

Sampson, S. 2015a. *Essentials of Service Design and Innovation.* 4th ed. Seattle. Washington: CreateSpace.

Sampson, S. 2015b. "Value Paradoxes and the Time Value of Value." *Service Science* 7, no. 3, pp. 149–62.

Sampson, S.E., and C.M. Froehle. 2006. "Foundations and Implications of a Proposed Unified Services Theory." *Production and Operations Management* 15, no. 2, pp.329–43.

Schuster, R., and R. Motal. July 20–23, 2009. "From e3-Value to REA: Modeling Multi-Party E-Business Collaborations." In *IEEE Conference on Commerce and Enterprise Computing, CEC 2009*. Vienna, Austria.

Shostack, G. 1984. "Designing Services that Deliver." *Harvard Business Review* 62, no. 1, pp. 133–39.

Spohrer, J., P.P. Maglio, J. Bailey, and D. Gruhl. 2007. "Steps Toward a Science of Service Systems." *IEEE Computer* 40, no. 1, pp. 71–77.

ter Hofstede, A.H.M., W. van der Aalst, M. Adams, and N. Russell, eds. 2010. *Modern Business Process Automation*. New York: Springer.

van Dinther, C., B. Blau, T. Conte, and C. Weinhardt. 2011. "Designing Auctions for Coordination in Service Networks." In *The Science of Service Systems*, eds. H. Demirkan, J.C. Spohrer, and V. Krishna, 107–129. New York: Springer.

Vandermerwe, S., and J. Rada. 1988. "Servitization of Business: Adding Value by Adding Services." *European Management Journal* 6, no. 4, pp. 314–24.

Vargo, S., and R. Lusch. 2004. "Evolving to a New Dominant Logic for Marketing." *Journal of Marketing* 68, no. 1, pp. 1–17.

Vargo, S., and M. Akaka. 2009. "Service-Dominant Logic as a Foundation for Service Science: Clarifications." *Service Science* 1, no. 1, pp. 32–41.

Vargo, S., H. Wieland, and M. Akaka. 2015. "Institutions in Innovation: A Service Ecosystems Perspective." *Industrial Marketing Management* 44, no. 1, pp. 63–72.

Von Neumann, J., and O. Morgenstern. 1944. *Theory of Games and Economic Behavior*. Princeton, NJ: Princeton University Press.

Wailgum, T. 2009. "10 Famous ERP Disasters, Dustups and Disappointments." http://www.cio.com/article/2429865/enterprise-resource-planning/10-famous-erp-disasters--dustups-and-disappointments.html (accessed June 24, 2015).

Wasserman, S., and K. Faust 1994. *Social Network Analysis*. Cambridge, UK: Cambridge University Press.

Wieland, H., F. Polese, S. Vargo, and R. Lusch. 2012. "Toward a Service (Eco) Systems Perspective on Value Creation." *International Journal of Service Science, Management, Engineering, and Technology* 3, no. 3, pp. 12–25.

Weske, M. 2007. *Business Process Management: Concepts, Languages, Architectures*. New York: Springer-Verlag.

Yu, E., P. Giorgini, N. Maiden, and J. Mylopoulos, eds. 2010. *Social Modeling for Requirements Engineering*. Boston. MA: MIT Press.

Index

Notes: Page numbers followed by "*f*" represents figures and "*t*" represents tables

OTHER TITLES IN OUR SERVICE SYSTEMS AND INNOVATIONS IN BUSINESS AND SOCIETY COLLECTION

Jim Spohrer, IBM and Haluk Demirkan, Arizona State University, Editors

- *Lean Sigma Methods and Tools for Service Organizations: The Story of a Cruise Line Transformation* by Jaideep Motwani, Rob Ptacek, and Richard Fleming
- *Designing Service Processes to Unlock Value* by Joy Field
- *Business Engineering and Service Design with Applications for Health Care Institutions* by Oscar Barros
- *Achieving Service Excellence: Maximizing Enterprise Performance Through Innovation and Technology* by Carl M. Chang
- *Service and Service Systems: Provider Challenges and Directions in Unsettled Times* by Steve Baron, Philip Hunter-Jones, and Gary Warnaby
- *Service Thinking: The Seven Principles to Discover Innovative Opportunities* by Hunter Hastings and Jeff Saperstein
- *Profiting From Services and Solutions: What Product-Centric Firms Need to Know* by Valarie A. Zeithaml, Stephen W. Brown and Mary Jo Bitner
- *People, Processes, Services, and Things: Using Services Innovation to Enable the Internet of Everything* by Hazim Dahir, Bil Dry, and Carlos Pignataro
- *Service Design and Delivery: How Design Thinking Can Innovate Business and Add Value to Society* by Toshiaki Kurokawa
- *All Services, All the Time: How Business Services Serve Your Business* by Doug McDavid

Announcing the Business Expert Press Digital Library

Concise e-books business students need for classroom and research

This book can also be purchased in an e-book collection by your library as

- a one-time purchase,
- that is owned forever,
- allows for simultaneous readers,
- has no restrictions on printing, and
- can be downloaded as PDFs from within the library community.

Our digital library collections are a great solution to beat the rising cost of textbooks. E-books can be loaded into their course management systems or onto students' e-book readers.
The **Business Expert Press** digital libraries are very affordable, with no obligation to buy in future years. For more information, please visit **www.businessexpertpress.com/librarians**. To set up a trial in the United States, please email **sales@businessexpertpress.com.**

Modeling Service Systems

Ralph D. Badinelli

This book invites the reader on a journey of discovery of service systems. From a Service-Dominant-Logic perspective, such systems are the building blocks of all economic activity, and innovation of new service systems holds the promise of a new industrial revolution. Users navigating web sites, customers interacting with intelligent mobile retail applications, patients interpreting advice from health-care professionals and other sources, students interacting with teachers and learning materials, city dwellers invoking smart service applications for transportation routing, and the unlimited variations of smart service systems that will be enabled by the Internet of Things and other technologies provide ample evidence of the need for service innovation.

This book presents an overview of the foundational constructs of service science and models of co-creative systems, with the aim of enabling the reader to be a service innovator. The value proposition of this book is the opportunity to fill each reader's knowledge gaps and offer a comprehensive, coherent, and introductory overview of service system modeling.

Ralph D. Badinelli is a chaired professor in the Department of Business Information Technology, Pamplin College of Business of Virginia Tech. He received PhD and MS degrees in management from the Krannert Graduate School of Business of the Purdue University, an MS degree in physics from the Purdue University, and a BS degree in mathematics and physics from the Hofstra University. Dr. Badinelli is a board member of the International Society for Service Innovation Professionals (ISSIP), an executive officer of the Service Research & Innovation Institute (SRII), former chairperson of the INFORMS Service Science Section, and a member of the Institute for Operations Research and Management Science (INFORMS), Project Management Institute (PMI), and the Association for Operations Management (APICS).

SERVICE SYSTEMS AND INNOVATIONS IN BUSINESS AND SOCIETY COLLECTION

Jim Spohrer and Haluk Demirkan, *Editors*

BEP BUSINESS EXPERT PRESS

ISBN: 978-1-63157-023-0

90000

9 781631 570230