Part I

Systems Thinking

Chapter **2**

Systems Thinking

PATRICK J. DRISCOLL, Ph.D.

> No man is an island entire of itself; every man is a piece of the continent, a part of the main.
>
> —John Donne (1572–1631)

> Is the world actually composed of interacting sets of entities that combine to create system? Or is the notion of a system simply a convenient way of describing the world? In the final analysis, we simply do not know.
>
> —Micheal Pidd [1]

2.1 INTRODUCTION

As his quote from *Meditation XVII* clearly illustrates above, the Irish poet John Donne was very much a systems thinker. He envisioned the human race as being strongly interconnected to the point that "any man's death diminishes [him], because [he is] involved with mankind." Had he been living today, he no doubt would have been a strong advocate of social network theory [2].

As one of the current champions for using a holistic approach when developing models of real-world phenomena, Michael Pidd poses two interesting questions above that go right to the heart of the discipline of systems engineering (SE). In a sense, whether systems are actual entities or simply one by-product of human

Decision Making in Systems Engineering and Management, Second Edition
Edited by Gregory S. Parnell, Patrick J. Driscoll, Dale L. Henderson

perception and reasoning is irrelevant. If systems are the natural means by which we cope with and understand the highly connected, information-intensive world we live in, it would seem illogical to not incorporate a strong consideration of the impact of this connectedness when making decisions about this same world. This is what systems thinking is all about.

> *Why system thinking matters*: How you think is how you act is how you are. The way you think creates the results you get. The most powerful way to improve the quality of your results is to improve the way you think [3].

Systems thinking in support of decision making consistently leads to deep understanding of most, if not all, of the various factors affecting possible alternatives. This success is largely due to two distinguishing characteristics: (a) the manner in which it departs from analytic thinking and (b) its natural ability to reveal subtle but important intricacies hidden to myopic or pure decomposition approaches that fail to consider "the big picture."

Applied to a systems decision problem, analytic thinking starts with the current system, identifies problems and issues that require fixing, applies focused modeling techniques to understand these deficiencies and identify possible solutions, and concludes with recommending solutions for changing some controllable dimensions of system activities that improve the system end state. Although the specific steps used in various disciplines may differ, this is the prevalent style of thinking imbedded in modern education. This way of thinking is most successful in situations where fine tuning of some system performance measures is called for but the system structure itself is assumed to be acceptable. These decision problems are often referred to as "well-structured." Improving the efficiency of established logistic supply networks, increasing the system reliability of telecommunication networks, and reducing transportation costs in package pickup and delivery systems are good examples of where analytic thinking has successfully supported decision making. In all these cases (and others), the *operation* of the system lies at the heart of the decision to be made and not the system itself.

In contrast, systems thinking first and foremost centers *on the system itself* [28, 29]. Operational improvements such as the ones noted above, representing only one dimension of the overall system structure, are identified as part of alternative system solutions crafted with stakeholder ideals clearly in mind. This style of thinking drives the systems decision process (SDP) introduced earlier.

For any system decision problem, system thinking starts with the system output ("What should the system do? What is desired by the stakeholders?") and proceeds to work backwards to identify system functions, processes, objectives, structure, and elements necessary to achieve this desired output. It then assesses the current state of the system ("Where is the system currently?") and asks, "What actions need to be taken to move the system from where it is to where it needs to be in order to maximize the value it delivers to stakeholders?" This natural focus on output (i.e., results, effects) provided by systems thinking creates a goal-oriented

frame of reference that produces long-term, effective *system-level solutions* rather than short-term, *symptom-level solutions*. This point bears emphasis.

Possessing the frame-of-reference afforded by systems thinking enables one to distinguish between symptom-level and system-level phenomena. Symptom-level phenomena are typically of short duration and easily observable. When they repeat, they tend to vary in character, intensity, and impact, thereby appearing as new phenomena to the untrained eye. Eliminating symptom-level problems provides short-term relief but will not prevent their recurrence in the future because the underlying system structure from which these symptoms arise is unchanged.

System-level phenomena are persistent, presenting themselves across a layer of commonality among all system components. These phenomena endure because they are an element or aspect of the underlying structure and organization of the system components, how these components interact, and the common ingredients that sustain their activity. System-level issues are identified using techniques that focus on identifying failure modes, such as root cause analysis [4] that attempt to trace collections of symptom-level effects back to shared sources of generation. System-level solutions provide long-term, fundamental system performance changes, some of which may not be predictable.

While symptom-level solutions can provide an immediate value return (e.g., stop crime in a neighborhood), they are not going to alter structural elements of the system that give rise to the observed symptom (e.g., cultural beliefs). Another way of saying this is that system-level solutions alter the fundamental system dynamics and relationships between system components; symptom-level solutions provide spot-fixes where these dynamics and relationships are failing. Risk-to-return on a specific investment instrument is a symptom-level phenomenon; elevated systemic risk shared across the entire derivatives market because of widespread use of collateralized debt obligations and credit default swaps is a system-level phenomenon [31]. A single company deciding to no longer participate in these financial products would provide localized risk-relief, but the underlying system-wide risk exposure still untreated will cause (possibly new) risk events to appear elsewhere.

It is possible in this framework that what are currently perceived as issues might not need fixing; they may actually be opportunities needing reinforcement. The perceived issues may very well be evidence of system functionality that is being imposed on the system by its users and is pushing against the constraints of the formally established structure and practices. In large organizations, "workarounds" created by employees in order to properly accomplish tasks and the emergence of informal leaders assuming ad hoc roles and responsibilities outside of the established hierarchy are often indicators of just such a situation. The stakeholder analysis so critical to the successful application of the SDP properly frames these issues within a broader system perspective as they arise without assuming they must be eliminated.

Adopting and maintaining a value focus in this setting is essential because as a system evolves through its life cycle, changing in size and complexity, the value being delivered by the system likewise changes. Sometimes this occurs in an undesirable way that drives system redesign efforts. Other times a more subtle

adaptation occurs in which the system shifts its value focus, obscuring from system owners and stakeholders exactly how and where it is delivering value. Systems thinking provides a frame through which to identify these subtle, but important, considerations.

> *Systems thinking* is a holistic mental framework and world view that recognizes a system as an entity first, as a whole, with its fit and relationship with its environment being primary concerns.

This philosophy underscores a systems engineering thought process predicated on the belief that the study of the whole should come before that of the parts, recognizing that there are system level behaviors, interactions, and structural characteristics that are not present when the system is decomposed into its elements. This sets apart systems engineering from classical engineering whose thought process is founded on the principle of decomposition as the basis of understanding. This philosophy has become indispensable when addressing modern systems whose size and complexity were not feasible less than a decade ago. Systems of systems engineering [5], model-oriented systems engineering [6], and techniques for designing complex systems [7] have emerged from the systems engineering community in response to this growing challenge. None of these approaches and their associated methods would exist in the absence of systems thinking.

The reason for departing from a pure decomposition principle at the onset of a systems decision problem is that decomposition is an activity that focuses on individual system element characteristics. It uses these individual characteristics to logically group or arrange elements so that the extent of shared characteristics becomes evident, thereby providing insights into how a more efficient or effective systems structure might be realized (by combining elements) or how a systems analysis might be more simply performed (because the analytical results associated with one element might apply to other elements possessing a high degree of shared characteristics with it).

Focusing on individual system elements tends to miss crucial interactions between the elements of a system or between composite groups of systems interacting as a whole. When these interactions or interdependencies are overlooked or insufficiently emphasized, the resulting modeling and analysis can suggest potential solutions that exhibit suboptimal characteristics. Such solution alternatives, while attractive to the performance of individual elements, can actually hinder or degrade some performance measure of the overall system. The risk (return volatility) for a portfolio of investments can easily increase because of an inappropriate over-investment in one particular asset, resulting in a loss of optimality for the portfolio [8]. In a similar fashion, installing high-intensity discharge (HID) headlamps into an older model vehicle may increase the lumens output (maximization effect for the headlamp element), but because the older model car is designed to take filament bulbs, doing so results in improperly focused beam patterns and excessive glare to other road users. A safety measure associated with the older vehicle as a transportation system would likely degrade as a result.

Systems have emergent properties that are not possessed by any of its individual elements. Bottlenecks and the "accordion effect" observed in dense highway traffic flows are examples of emergent properties not possessed by individual automobiles; they become evident only when the transportation elements are viewed as a whole system. These properties result from the relationships between system elements, commonly described as the system *structure* (Figure 2.1). In many cases, this structure can be described mathematically. The functions and expressions resulting from these mathematical descriptions directly support modeling the system as described in Chapter 4.

Systems thinking enables one to progress beyond simply seeing isolated events to recognizing patterns of interactions and the underlying structures that are responsible for them [9]. It reveals the structure in a way that enables us to specify the boundary of the system, which sets apart the system and its internal functions from the environment external to it. Knowing this boundary enables us to identify key system inputs and outputs and to visualize the *spatial arrangement* of the system within its environment. Critical systems thinking of the type required for systems engineering encourages creativity simply because of the strong interplay between conceptual visualization, detailed analysis, and unique measures of effectiveness produced by synthesizing ideas.

The combination of systems thinking with best engineering practices has produced a variation of systems engineering second to none. The particular application

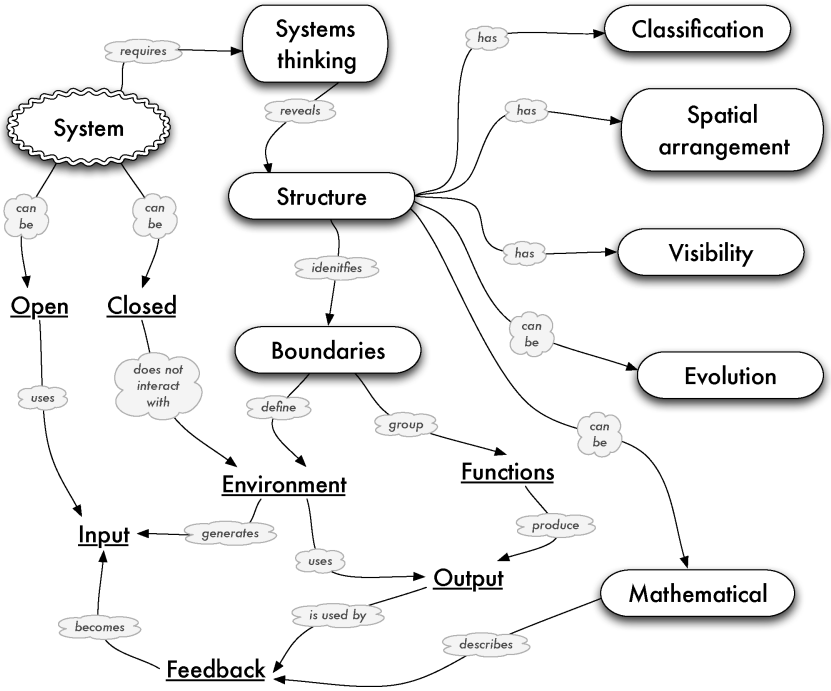

Figure 2.1 Concept map for Chapter 2.

of systems thinking advocated in this book is an adaption of general systems theory [10] with an emphasis on systems decision problems.

2.2 STRUCTURE

Understanding the workings of a system requires some conceptual thinking about its underlying *structure*. For systems engineering, this means several things: identifying the elements of the system, understanding how the elements are connected and how they interact with each other to achieve the system's purpose, and understanding where this system lies relative to other systems that can impact on its behavior or vice versa.

Different systems afford different levels of visibility into their inner workings. The relationships between elements can change dynamically over time, causing system evolution to become a concern. A portion of the output of a system can undergo *feedback* in which the environment uses the system output to create conditioned input that is fed back into the system. Interaction with the environment requires that a system be *open*, because a *closed* system has no interaction across its boundaries.

Consider for a moment the decision associated with purchasing an automobile. On the one hand, the decision might appear to be a simply structured one: go to a local dealership, examine their inventory, compare prices to budget, and purchase the vehicle that comes closest to satisfying desire without violating the working budget limits. No doubt many people exercise this strategy on a routine basis. Knowing this propensity of potential buyers enables automobile manufacturers to craft enticing advertisements and marketing campaigns that can effectively reshape what a potential buyer thinks they want into what the manufacturer wants to sell. This principle lies at the heart of effective marketing [11].

On the other hand, applying a small amount of systems thinking to the car buying decision reveals the purchase decision as highly connected in its structure. The car being purchased will become part of a system in which the automobile is a new element that interacts in various ways with other major systems: the transportation highway system (health & safety), banking (financial) systems, fuel logistic (technical) systems, personal prestige and entertainment (social) systems, insurance (financial) systems, motor vehicle regulation (legal) systems, communication (technical) systems, ecological systems, and so on. From this systems thinking perspective, the purchase decision takes on a much greater level of importance than it may have otherwise. It clearly has an impact on each of these other systems in some manner that should be either taken into consideration or intentionally disregarded, but certainly not ignored.

In fact, one of the most significant failings of the current U.S. transportation system is that the automobile was never thought of as being part of a system until recently. It was developed and introduced during a period that envisioned the automobile as a stand-alone technology largely replacing the horse and carriage. As long as it outperformed the previous equine technology, it was considered a success. This success is not nearly so apparent if the automobile is examined from a

systems thinking perspective. In that guise, it has managed to fail miserably across a host of dimensions. Many of these can be observed in any major U.S. city today: oversized cars and trucks negotiating tight roads and streets, bridges and tunnels incapable of handling daily traffic density, insufficient parking, poor air quality induced in areas where regional air circulation geography restricts free flow of wind, a distribution of the working population to suburban locations necessitating automobile transportation, and so on. Had the automobile been developed as a multilateral system interconnected with urban (and rural) transportation networks and environmental systems, cities would be in a much different situation than they find themselves in today.

What is important here is not that the automobile could have been developed differently, but that in choosing to design, develop, and deploy the automobile as a stand-alone technology, a host of complementary transportation solutions to replace the horse and buggy were not considered. Systems thinking would have helped to identify these potentially feasible solutions. If they were subsequently rejected, it would have been for logically defendable reasons directly related to stakeholder requirements. In the business of supporting decision making, limiting the span of potential solutions tends to degrade the quality of the chosen solution, certainly against a criteria of robustness.

An example in a social setting can further illustrate this point. In the United States during the late 1960s, it was not uncommon to hear people taking positions on issues of behavior choices by saying, "Why should it matter to anyone else what I do? If I decide to do this, I am the only one affected. It's my life." While choice certainly is within an individual's control, this statement ignores any and all connections and interactions between the individual expressing this position and subsystems of the metasystem within which the individual lives. John Donne recognized the importance of these connections nearly four centuries earlier, as evidenced by his quote at the start of this chapter.

2.3 CLASSIFICATION

Expanding the way we think about challenging systems decisions requires a top-down classification scheme that starts with the "big picture" of the system and its observable behavior. As stated earlier, we formally define a system as follows:

> A *system* is an integrated set of elements that accomplish a defined objective.

Systems occur in many different forms. We generally use three classes for describing the various system types: physical, abstract, and unperceivable. A *physical* system, also referred to as a concrete system [10], exists in the reality of space–time and consists of at least two elements that interact in a meaningful manner. This is a system that is clearly evident in the real world and directly observable to the trained and perhaps untrained eye. An automobile is a good example of a physical system.

Physical systems further subdivide into four subclasses that can overlap in some cases:

- *Nonliving* has no genetic connections in the system and no processes that qualitatively transform the elements together with the whole and continuously renew these same elements.
- *Living*, typically referred to as an organic system, is a system subject to the principles of natural selection such as an ant colony, a predator–prey relationship between species, and human biophysiology.
- *Manmade* physical systems intentionally created to augment human life such as transportation road networks, logistic resupply systems, and communication systems.
- *Natural* are those systems coming into being by natural processes, such as waterway networks created by natural processes associated with glaciers, tectonic plate shifts, and weather.

An *abstract* system is a system of concepts that are linked together in some manner, generally in an attempt to convey an initial design, a strategic policy, or some other idea that has not been implemented in some other form. Abstract systems are organizations of ideas expressed in symbolic form that can take the form of words, numbers, images, figures, or other symbols. In a sense, this type of system is an intermediate system pinched between reality and the completely intangible as its elements may or may not be empirically observable, because they are relationships abstracted from a particular interest or theoretical point of view.

Figure 2.2 is an example of the field of engineering management (EM) expressed as an abstract system. In its organization, the diagram conveys the order and purpose of the professional field of EM as emerging from the unique interaction of four separate disciplines: leadership, management, economics, and engineering. Each of these four disciplines could in turn be represented as systems. Permeating all of these systems are the environmental resource considerations of people, technology, time, and finances. A similar illustration could be used to show how the car buying decision (substituted in place of "EM" in the picture) impacts the other systems mentioned earlier.

An interesting point to note with regard to this figure is that the EM system as illustrated does not exist apart from the four discipline systems shown. It exists solely because of the interaction of these four systems. It is, in fact, a professional field delivered only at a holistic level; it is not possible to decompose the abstract EM system representation into its multilateral system elements and still retain the complete character of EM. Bear this observation in mind when the concept of designing "system-level measures of performance" arises later in this book.

An *unperceivable* system is a classification that is largely based on the limitations of our ability to observe the system rather than some innate characteristics it may possess. These systems exist when an extreme number of elements and the complexity of their relationships mask the underlying system structure or organization. A system representation used to describe an unperceivable system is an

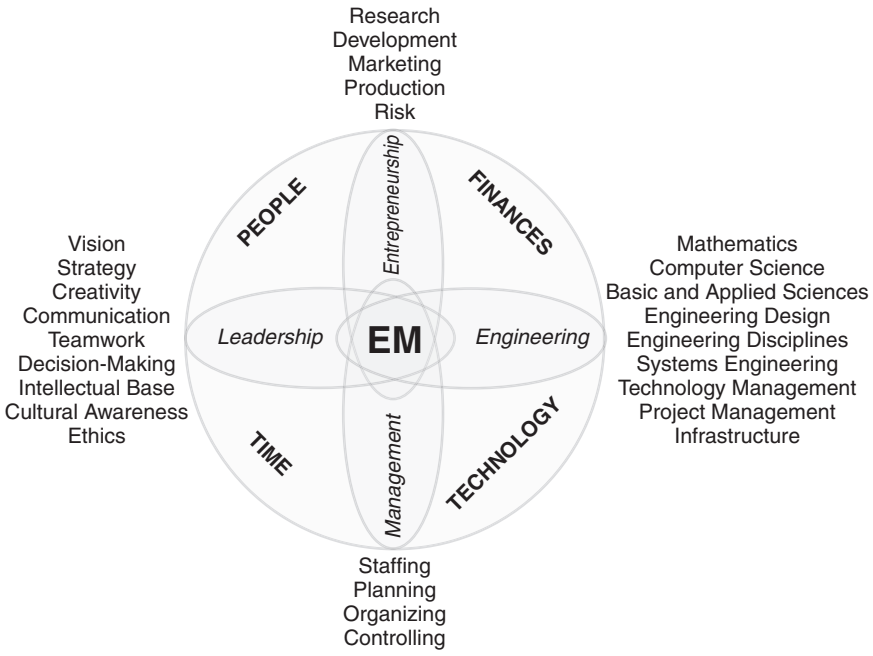

Figure 2.2 Conceptualization of engineering management system. (Courtesy of Dr. John Farr, U.S. Military Academy.)

approximation of the system at best, typically containing only as much detail so as to enable its inputs and outputs to be identified.

An example of an unperceivable system is the U.S. economy. A complete description that includes all of its elements and interrelationships defies our comprehension. At best, we resort to surrogate measures such as gross domestic product (GDP), the Dow Jones composite index, unemployment estimates, inflation rate estimates, foreign trade balance, monetary exchange rates, and new housing starts as indicators of how well (or poorly) the U.S. economy is performing. The complex nature of this system due to the number of elements, number of interactions, and evolving states of both these make it unperceivable as a system. This goes a long way toward explaining why, despite technology advances and Nobel laureate awardees in economics, error-free future state forecasts of this system remain impossible to attain.

2.4 BOUNDARIES

The concept of establishing a *system boundary* is fundamental to working with systems. From a practical standpoint, a system boundary serves to delineate those elements, interactions, and subsystems we believe should be part of a system definition from those we see as separate. To a good extent, what is included within

a system boundary is influenced by the systems decision motivating the analysis. Unfortunately, this means that there is no easy answer to setting a system boundary; every systems decision problem motivates its own appropriate system boundary and system definition. Systems engineers accomplish this using the information resulting from research and extensive interaction with stakeholders during the problem definition phase of the SDP. In this regard, the system boundary is one tool available to a systems engineer to help define the scope of a particular project.

A *system boundary* is a physical or conceptual boundary that encapsulates all the essential elements, subsystems, and interactions necessary to address a systems decision problem. The system boundary effectively and completely isolates the system under study from its external environment except for inputs and outputs that are allowed to move across the system boundary.

A system's boundary distinguishes the system from its environment. *Open* systems interact with their environment in a specific way in which they accept inputs from the environment in order to produce outputs that return to the environment. *Closed* systems are isolated and hermetic, accepting no inputs beyond those used to initialize the system and providing no outputs to its environment. Closed systems do not need to interact with their environment to maintain their existence. The clearest example of a closed system is one associated with physics and mechanical engineering: a perpetual motion device. Once initial energy is provided to put the device in motion, it stays in motion forever with absolutely no inputs from outside of its boundary.

Closed systems generally do not occur in nature. More often, human intervention in the form of controlled scientific experiments create closed systems. Some systems are considered closed systems because some particular exchange across its boundary is discounted or ignored. Atoms and molecules can be considered closed systems if their quantum effects are ignored. Greenhouses could be considered closed systems if energy (heat) exchange is ignored. In mathematics, an abstract system, vector spaces are closed systems: Valid operations performed on the elements of a vector space remain in the vector space forever. Edwin A. Abbott (1838–1926), an English schoolmaster and theologian, wrote an interesting book titled *Flatland: A Romance of Many Dimensions* [12] about what life would be like in a two-dimensional world. He had this closed system idea in mind.

Most, if not all, systems we encounter and operate in our daily lives are open systems. Consequently, if an initial inspection of a system makes it appear to be a closed system, chances are we are overlooking some multilateral or hierarchical systems that are affecting input to the system.

The boundary of a system is the actual limits of the major elements. Inputs and outputs cross boundaries of systems. The boundary of a system must be selected to include all of the important interacting elements of the system of interest and

exclude all those that do not impact the system behavior that makes it a system. Isolating the core system is an important part of a systems engineering project because doing so allows lateral systems, subsystems, and metasystems to be identified as well.

Figure 2.3 illustrates a hypothetical open system with each of the major system structural elements shown. The system accepts input from its environment, acts on that input via its internal functions, and again interacts with its environment by producing output. Input can take the form of physical material, as in sand being used as a source of silicon dioxide for making silicon wafers for computer microchips. Input can also possess a less tangible form, as in the case of information. For decision support systems used to predict stock performance or assign airline ticket prices, information is one input flowing across the boundary of these open systems.

While it may be natural to think of these inputs and outputs in terms of matter, energy, and materials such as raw materials used in a manufacturing or production facility, they can just as well be services, or induced effects such as influences and other psychological entities. These can all be appropriately defined as crossing a particular system boundary. The strength of this conceptualization predominately lies in its broad application across a very wide spectrum of disciplines and applications.

Figure 2.3 also illustrates the two major versions of system *feedback*: internal and external [13].

> *Internal* feedback is the feedback of a system that is modified and recycled with the system boundary to alter inputs delivered to a system.

Internal feedback is entirely controlled by the system and is typically not visible from outside the system boundary. A common example of internal feedback is a manufacturing quality control process that sends work-in-progress back through manufacturing stages for rework prior to releasing the product for

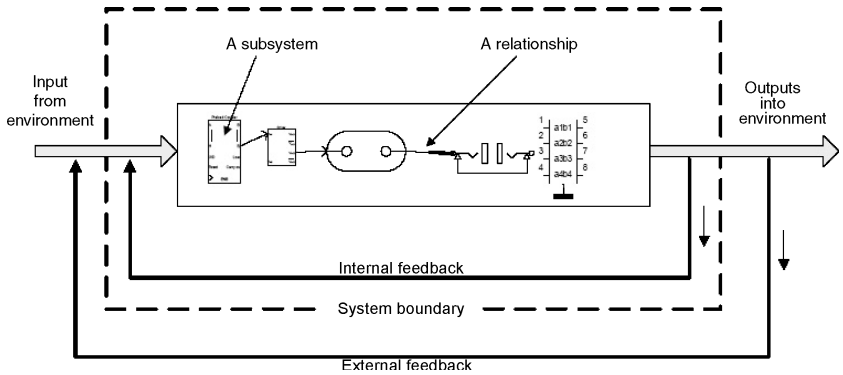

Figure 2.3 Structural organization of a system with boundaries.

distribution. A consumer only sees the end product, not being privy to the internal workings of the manufacturing company. In education, intradepartmental reviews of courses, syllabi, programs, and teaching philosophies are other examples of internal feedback if a system boundary for an academic major were drawn around a department's operations.

Identifying internal feedback depends on having access to the inner workings of a system. If the owner of a system allows access to its internal processes and structure, then it is possible to identify internal feedback. Otherwise, internal feedback is invisible to external observation. When a Securities and Exchange Commission (SEC) accountant performs an in-depth financial audit of a publicly traded company in the United States, full access is required so that the inner workings of the company (aka: systems internal processes) can be identified and examined for adherence to SEC regulations. In a similar fashion, ABET, Inc. requires such access to university academic programs in order to accredit systems engineering major programs.

External feedback is the feedback of a system that occurs in the environment outside of the system boundary, acting to alter inputs before they are delivered to a system.

A system "sees" external feedback in the form of its normal input. Without some means of external observation, a system is unaware of the external systems processes and interactions that are using or responding to some portion of its own output, integrating this output into their systems functions, and releasing modified outputs into the environment that becomes part of the system inputs. This underscores a very important point as to why systems engineers add value to customer programs even when the customer is extremely talented and knowledgeable at what they do: It always helps to have a fresh set of eyes on a problem if for no other reason than to provide objective clarity on systems operations.

Feedback complicates systems modeling and analysis. Modern systems, which are highly connected to other systems both to survive competition and to leverage cooperation, have designed processes enabling them to adapt to their surroundings as a means of attaining competitive advantage and improving performance. Systems that modify their output in a manner that actively responds to changes in their environment due to injection of other systems output are called *adaptive* systems. Adaptive systems tend to pose more of a modeling and analysis challenge because the external (and possibly internal) feedback pathways need to be identified in the course of understanding how the system is operating.

It is also possible that while some open systems produce outputs that are simply transformed input as shown in Figure 2.3, this is not always the case. It is possible that the outputs of the system are simply there to allow the system to secure, through a cycle of events, more of the useful inputs it needs to survive.

In the conceptual system of Figure 2.2, the apparent boundary of the engineering management discipline would be the perimeter of the central, dark circle in the figure which sets it apart from the other overlapping systems.

2.5 VISIBILITY

In most broad terms, system structure can be described first in terms of its relationship with the environment and secondly in terms of what it represents and how it is organized to interact. We have seen one form of structure already: open or closed systems that capture how the system interacts with its environment. Here we want to examine two other dimensions of system structure: visibility and mathematical.

From the systems perspective of input, transform, and output, we can describe system structure in terms of the degree of visibility that we have on the internal workings of the system. Are the elements and their interactions readily identifiable, or are some or all of these hidden from view? The answer to this question enables us to specify the system as one of the three basic structural models: a black box, a gray box, or a white box [10]. These three gradations of visibility into the inner workings of a system are illustrated in Figure 2.4.

A *black* box is a structure that behaves in a certain way without providing any visibility into exactly what internal elements are, how they are specifically linked, and how they functionally transform inputs to the system to produce observable system output. Uncovering such information for the purposes of gaining deeper understanding of a black box system consists primarily of repeatedly manipulating

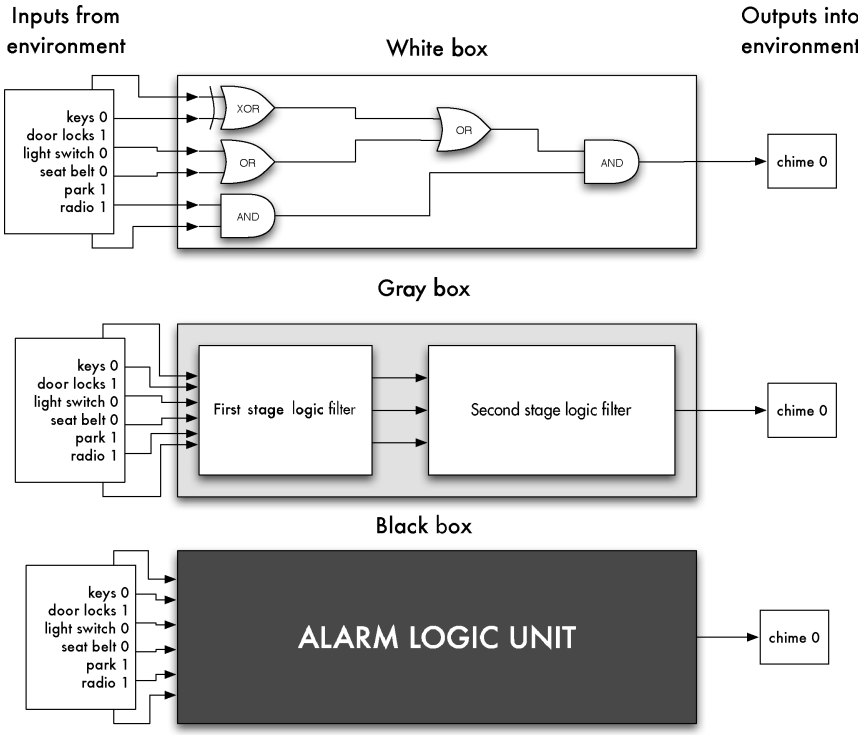

Figure 2.4 Degrees of internal understanding of a system.

the input to get to a point of producing reasonably predictable output(s). Extracting the functional linkage between input and output then becomes the task of design of experiments, regression, response surface methodology, and other data-dependent modeling techniques (see Chapter 4).

A good example of a black box system is human body temperature regulation. If a person's body temperature is too high, placing that person in a bath of cold water lowers it, as does administering loading doses of aspirin. If the person's body temperature is too low, slow heating in a hot tub or with human skin contact will gradually raise the temperature. How exactly the body is reacting to these external stimuli at a subsystem interaction level to make this happen is known, but generally not accessible to discovery without extremely sophisticated instrumentation.

A *gray* box offers partial knowledge of selected internal component interactions and processes (activities). This middle ground perspective is one that is often encountered in practice because even very complex systems have partially accessible and understandable internal processes. For example, in the U.S. economic system which is unperceivable, we could create groups of elements at various levels that would allow a partial view into the inner workings of this system, such as country elements engaged in foreign trade, all legal nonprofit organizations engaged in charity work in the state of New York, and so on. In the gray box illustration of Figure 2.4, while a select number of system elements are visible, their interactions are not. Complete information concerning the system this represents cannot be acquired.

A *white* box systems perspective recognizes complete transparency on a systems internal elements and processes. This ability is rarely achievable when working with existing systems that are complex, but it is very common in newly designed systems such as telecom networks, mechanical devices, business partnerships, athletic teams, and farming operations. It is possible to reasonably fill in gaps in understanding of internal elements and processes of white box systems by making assumptions and then validating these assumptions by measuring output variation in comparison to input changes, as would be done with a black or gray box system.

2.6 IDEF0 MODELS

Black box representations serve a useful purpose early on in the process of designing a system, when system concepts are being explored and requirements are being identified based on system needs and major functions. At this stage, conceptual tools leveraging system thinking are helpful. As will be seen in Chapter 10, once the desired critical functions for achieving the overall systems purpose have been identified, a black box representation for these functions can be created using a method developed by and made public in 1981 by the U.S. Air Force Program for Integrated Computer-Aided Manufacturing (ICAM). More recently, the U.S. Department of Commerce issued Federal Information Processing Standards (FIPS) Publication 183 that defines the Integration Definition for Function Modeling (IDEF0) language, a formal method of describing systems, processes, and their activities.

TABLE 2.1 Family of IDEF Modeling Methods [14]

Method	Purpose	Method	Purpose
IDEF0	Function modeling	IDEF8	User interface modeling
IDEF1	Information modeling	IDEF9	Scenario-driven IS design
IDEF1X	Data modeling	IDEF10	Implementation architecture modeling
IDEF2	Simulation model design	IDEF11	Information artifact modeling
IDEF3	Process description capture	IDEF12	Organization modeling
IDEF4	Object-oriented design	IDEF13	Three schema mapping design
IDEF5	Ontology description capture	IDEF14	Network design
IDEF6	Design rationale capture		

IDEF0 is one member of a family of IDEF methods [15] that can be used to support the SDP during the early phases of a system design effort. IDEF0 models describe the functions that are performed by a system and what is needed to perform those functions. Table 2.1 shows the variety of methods and the purpose for which they were designed [14].

An IDEF0 model is capable of representing the functions, decisions, processes, and activities of an organization or system. It is particularly useful for representing complex systems comprised of a host of processes. The top-level diagram, called Level 0, is the highest level of system abstraction. Using a single box to represent the top level system function, it illustrates in equally high level terms the things that cross the system boundary as inputs and outputs, the physical mechanisms that enable the top level system function, and the controls that determine how this function will operate. Levels 1, 2, and so on, proceed to decompose this Level 0 representation, successively exposing more and more detail concerning the interconnections and interactions of the various subfunctions supporting the overall system function. Proceeding from Level 0 to Level 1, for example, is a bit like lifting the lid off of the black box representation for a system; we begin to see the sequence of processes, activities, or functions linked together to successfully perform the top level function. Even though IDEF0 models are generated using a decomposition strategy, they tend to maintain a "big picture" orientation to the desired system that other approaches such as functional flowcharts can lose as modeling detail increases. It is particularly useful when establishing the scope of a functional analysis as it forces the user to decide on the system boundary in order to display even the highest level of representation for a system.

A process is a systematic series of activities directed toward achieving a goal. IDEF0 uses a single box to represent each activity, combines activities to define a process, and then links system processes to describe a complete system. Creating IDEF0 models of a system generally proceeds by decomposing system functions into layers, the top layer being a system-level model resembling a black box based on the information obtained during stakeholder interviews. In this sense then, it is helpful to decide whether an IDEF0 model is going to be used prior to interviewing

Figure 2.5 Example SIPOC diagram for a community policing process.

stakeholders. Doing so ensures that key information required for the IDEF0 model can be obtained without having to unnecessarily re-interview stakeholders.

IDEF0 represention are similar in structure to the popular SIPOC diagrams commonly used as an intial step taken when mapping processes for Lean Six Sigma applications [16]. SIPOC is an acronym for Suppliers, Input, Process, Output, and Customers. Figure 2.5 illustrates a SIPOC diagram for a hypothetical service process provided by a community-based law enforcement organization.

What differs is the particular level of detail used in the representation. SIPOC diagrams graphically illustrate a simple flow of service or products between suppliers and customers as a first step toward a more detailed value stream mapping that contains process controls and mechanisms along with a host of more details necessary to identify non-value-added components of the process that are then targeted for elimination. Turtle diagrams [17], a far less popular modification of SIPOC diagrams, include information "legs" to the SIPOC diagram that contain details concerning measures, what, how, and who about the process.

Creating an IDEF0 model loosely follows a stepwise procedure. At the start, the overall purpose for the model and the particular stakeholder(s) viewpoint that is going to be reflected in the IDEF0 must be identified on the Level 0 representation. This explicit statement of viewpoint orients a user of the IDEF0 model as to the perspective that should be assumed when interpreting the information in the model. Next, using both stakeholder interviews and independent research, identify the system, its boundary, major raw material inputs and outputs, the top level system function that turns inputs into outputs, and the laws, regulations, operating guidelines, and stakeholder desires that control this function, along with the physical aspects that cause the system to operate. This information will become the top level IDEF0 model.

Following this, identify the major processes that are needed to turn the system inputs into outputs. For each major process, identify the objectives or purpose associated with it. These processes typically are sequentially linked at the highest level. Each of these processes will further break down into sequences of activities organized to achieve the objective(s) of the process. All three of these system structures, the system, the processes, and the activities, are represented as individual IDEF0 models.

Lastly, it is important to decide on the decomposition strategy that will be used. Four common decomposition strategies are [18] as follows:

- *Functional decomposition* breaks things down according to *what* is done, rather than *how* it is done. This tends to be the most common decomposition

strategy because it naturally aligns with the strategy used for constructing a functional hierarchy.

- *Role decomposition* breaks things down according to *who* does what.
- *Subsystems decomposition* starts by breaking up the overall system into its major subsystems. If the major subsystems are relatively independent, this is a helpful strategy to initiate the IDEF0, subsequently employing functional decomposition to further decompose the subsystems into processes.
- *Life cycle decomposition* is sometimes used when the stages of the system life cycle are relatively independent and subsystems and their processes generally align with the age of the system.

At the end of these steps, an initial IDEF0 system representation can be formally constructed. From a practical standpoint, an IDEF0 model is a conceptualization that supports functional analysis and ultimately develops a value hierarchy within the SDP. IDEF0 revisions are commonplace during the SDP as new and more accurate information arises with regard to the needs, wants, and desires of stakeholders.

A basic IDEF0 model at any level consists of five possible components [18]:

1. *Activity, Process, or System.* A box labeled by "verb–noun" describing the activity/function that the box represents (e.g., collect intelligence; weld joint; coffee making).
2. *Inputs.* Arrows entering the left side of the box represent the "raw material" that gets transformed or consumed by the activity/function in order to produce outputs (e.g., information; welding rod, electric current, Tungsten Inert Gas (TIG); coffee grounds, water).
3. *Controls.* Arrows entering the top of the box are controls. These specify the conditions required for the function/activity to produce outputs, such as guidelines, plans, or standards that influence or direct how the activity works (e.g., U.S. federal regulations; union safety standards; recipe, coffee machine directions).
4. *Mechanisms.* Arrows connected to the bottom side of the box represent the physical aspects of the activity/function that cause it to operate (e.g., agents; union welder, TIG welding torch, electricity; drip coffee machine, coffee person, electricity). These can point inward or outward of the box. Inward pointing arrows identify some of the means that support the execution of the activity/function. Arrows pointing outward are *call* arrows. These enable the sharing of detail between models or between portions of the same model.
5. *Outputs.* Arrows leaving the box on the right are outputs, which are the result(s) of the activity/function transmitted to other activities/functions within the IDEF0 model, to other models within the system, or across the system boundary to the environment (e.g., intelligence reports, fused metallic bond, pot of coffee).

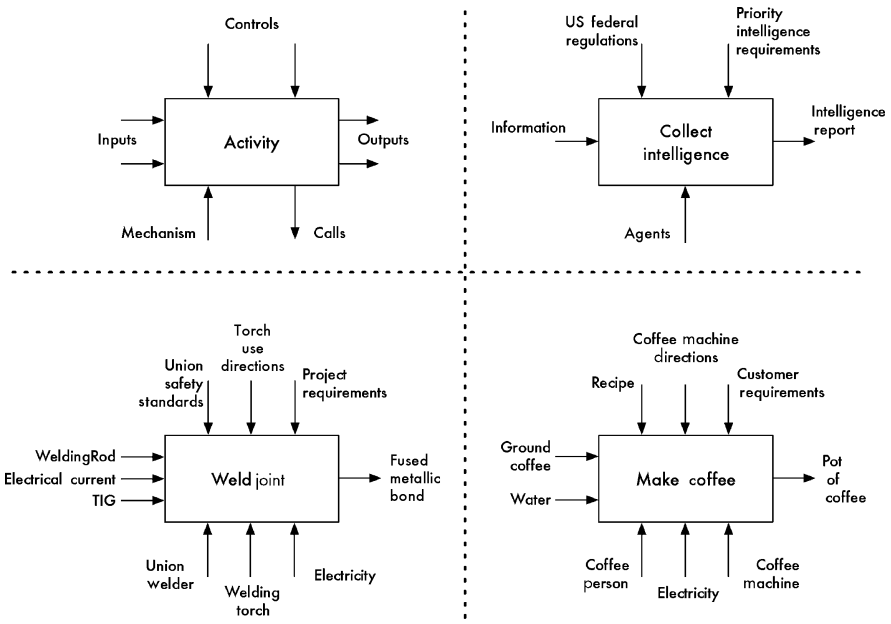

Figure 2.6 Generic IDEF0 model with three functional examples.

The upper left illustration in Figure 2.6 shows the generic structure of an IDEF0 model. The three other models are examples of IDEF0 models for each of the example activities noted in the description above. Constructing even the Level 0 representation of a system can be challenging. Referring to the upper left illustration in Figure 2.6, it is helpful when building each block in a diagram to conceptualize the block as a function that "transforms inputs 1 and 2 into outputs 1 and 2, as determined by controls 1 and 2 using mechanism 1." The verbs in this expression add clarity to the definitions of inputs, outputs, controls, and mechanisms.

Creating an IDEF0 model of a system proceeds in a decomposition fashion similar to that used to create a hierarchy. The highest level model of the system, Level 0, communicates the system boundary in relation to its environment. This top-level representation on which the subject of the model is represented by a single box with its bounding arrows that is labeled as A-0 (pronounced "A minus 0"). The A-0 diagram at Level 0 sets the model scope, boundary, and orientation through the visualization of the box, the purpose and viewpoint expressed in text below the box, and the title of the A-0 diagram at Level 0. The title affords the opportunity to identify the name of the system. The purpose communicates to a reader why they are looking at the IDEF0 model, and whose viewpoint it represents. Remember that a fundamental assumption of system thinking is that perspective matters. If the perspective of a system changes, that is, we look at the system through someone else's eyes, the system might look different. Again, this is precisely why multiple stakeholders must be included in any systems study.

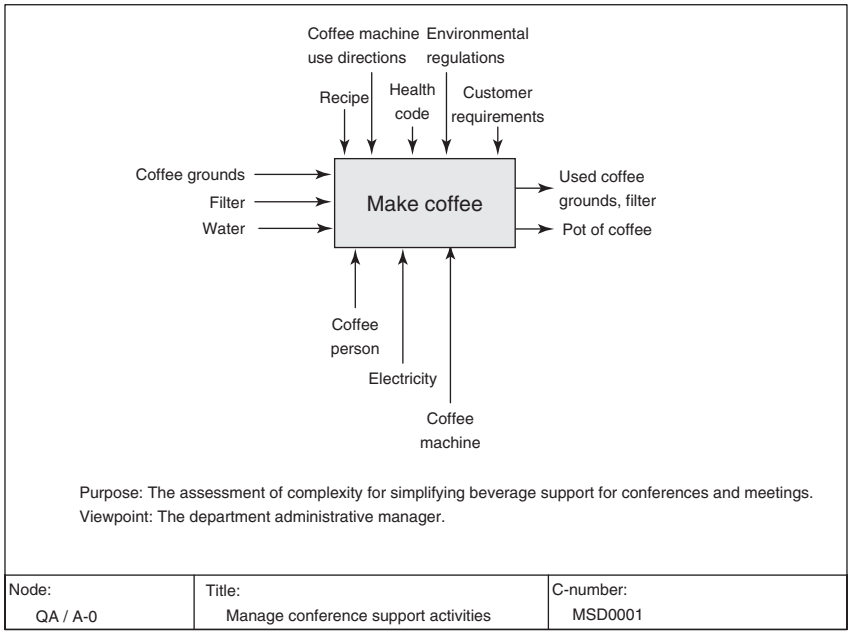

Figure 2.7 An A-0 diagram for the make coffee function.

Figure 2.7 shows a slightly more complicated A-0 diagram for the "Make Coffee" function introduced in Figure 2.6. The boundary of the system, purpose, and viewpoint are shown in the main diagram area. Along the bottom of the A-0 diagram are tracking information specified by the systems engineer who would create the diagram. The "ProjectID" (QA) and "NodeID" (A-0) are unique identifiers. The "Title" block is used to describe the overall activity in the context of the process, subsystem, or system it supports. The "C-number" is a combination of the author's initials (MSD) and document ID number (0001).

We chose this example for three reasons. First, notice that this Level 0 model assumes that a particular mechanism (electric coffee machine) is to be used in any systems solution that might result, which appears reasonable given the viewpoint expressed. However, for most applications of the SDP, we would caution to avoid including solution elements (how to do something) early in the Problem Definition phase, which is when the IDEF0 representation is most likely to be used. Coffee can be made without either electricity or a coffee machine. If the department administrative manager placed a high preference (value) on some functionality provided by an electronic coffee maker, it would be better to include this as stakeholder input for the next level of decomposition (Level 1) rather than as a hard mechanism at this modeling level (Level 0). In this way, solution alternatives are free to creatively satisfy this desired functionality using mechanisms other than an electric coffee maker, if feasible.

Second, even a simple process of making coffee for a meeting can take on complex interactions when viewed from a systems thinking perspective, and the

SDP has useful cues for where to look to recognize these interactions. For example, health code and environmental regulations (legal factor) exert a nontrivial amount of control over this function when the output is to be served in a public forum. Thus, the Make Coffee function interacts with the legal system.

Finally, comparing the output of this IDEF0 model with that of Figure 2.6, notice that we have included the waste products (the unintended consequences) along with the designed product (the intended consequences) as output of the system. In this way, the Make Coffee function as a system interacts with the environmental system existing outside of its system boundary. This interaction is easy to overlook, as some industrial companies operating in the United States have learned the hard way. (See http://www.epa.gov/hudson/)

Figure 2.8 illustrates an A-0 diagram for the SDP introduced in this book. The model shows a top-level view of the SDP, which accepts the current system status as input and produces the desired end-state system solution as output. Stakeholder input and the applicable systems engineering professional standards act as controls for the process execution. The systems engineering project team and organization resources comprise the physical aspects of the SDP.

An IDEF0 model of the SDP then proceeds to decompose the process into the next level, Level 1 (labeled as A-1) of system processes, namely the four phases of the SDP. Figure 2.9 shows a representation at Level 2 of the Problem Definition phase of the SDP. This Level 2 model displays the linkage of the three major SDP

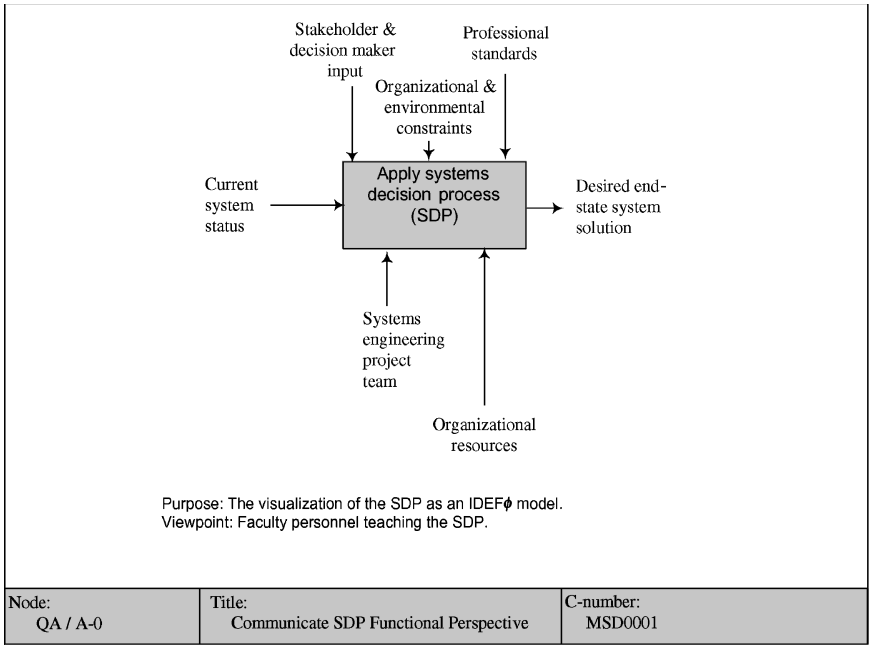

Figure 2.8 An A-0 diagram for the systems decision process (SDP).

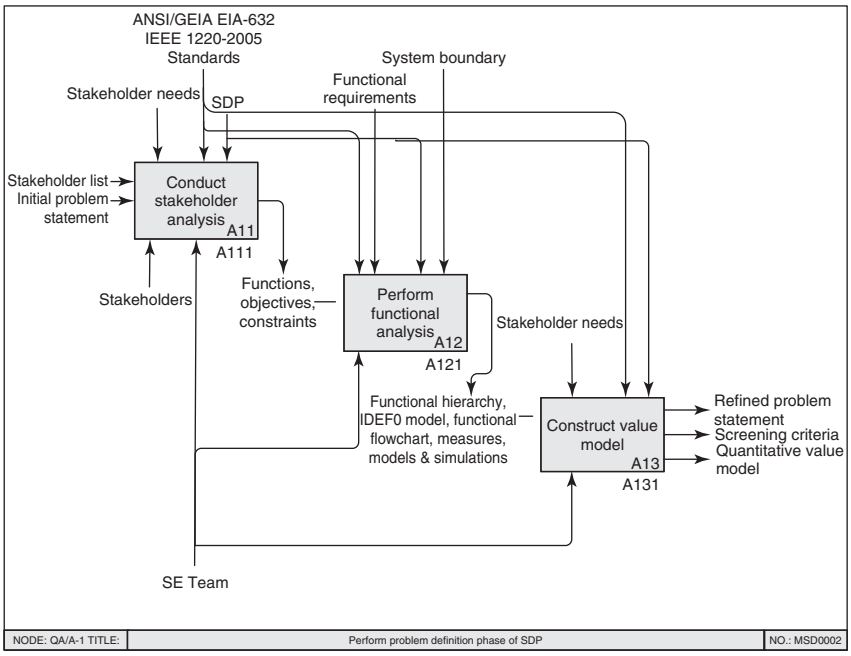

Figure 2.9 Level 2 representation of the problem definition phase of the SDP.

functions performed during this phase. Starting with a stakeholder list and initial problem statement as input, the three system functions (activities, processes) flow from the upper left to the lower right of the Level 2 model following the order in which inputs are handled to create the three necessary outputs required by the decision gate that allows the SDP to progress to the Solution Design phase that follows.

Although sequenced IDEF0 models like this one are commonly linked through inputs and outputs, they can also be linked by controls and mechanisms as well. In Figure 2.9, the professional systems engineering standards (ANSI/GEIA EIA-632 and IEEE 1220-2005) and the SDP act as control on each of the functions shown. Likewise, the SE team links the three functions by acting as a common physical mechanism enabling each of the functions to successfully occur. Each of the subsequent SDP phases can be represented by an IDEF0 diagram as well, which is left as an exercise for the reader.

Figures 2.10, 2.11, and 2.12 illustrate an example of what three levels of IDEF0 models could look like for a hypothetical fast food restaurant in the United States. The Level 1 model in Figure 2.11 presents the top-level system function as a sequence of three subfunctions: “process customer order,” “prepare food items,” and “deliver food items.” Figure 2.12 then shows how the “process customer order” function is decomposed into a sequence of four functions. In turn at Level 2, similar models would be created for the “prepare food items” and “deliver food items”

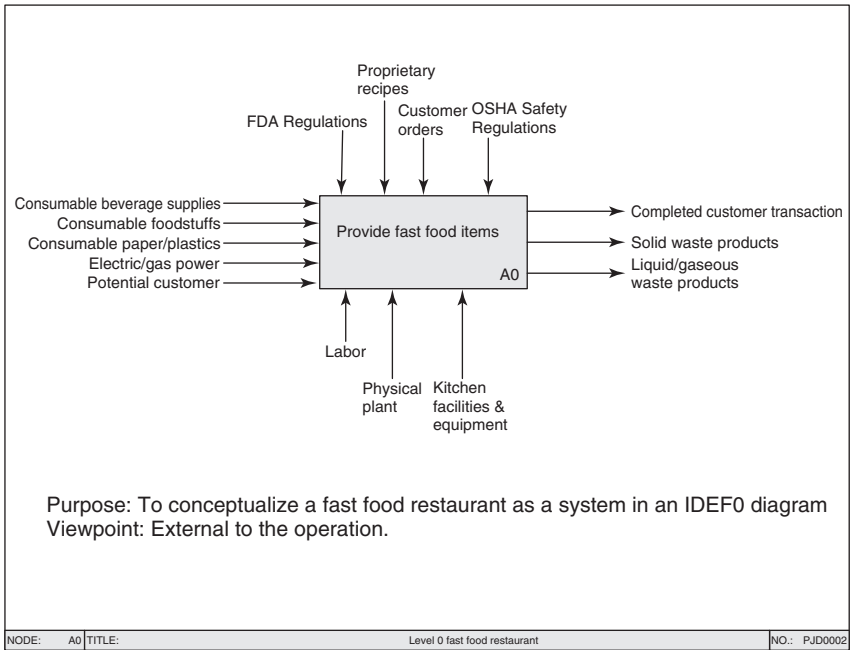

Figure 2.10 Level 0 model of a fast food restaurant.

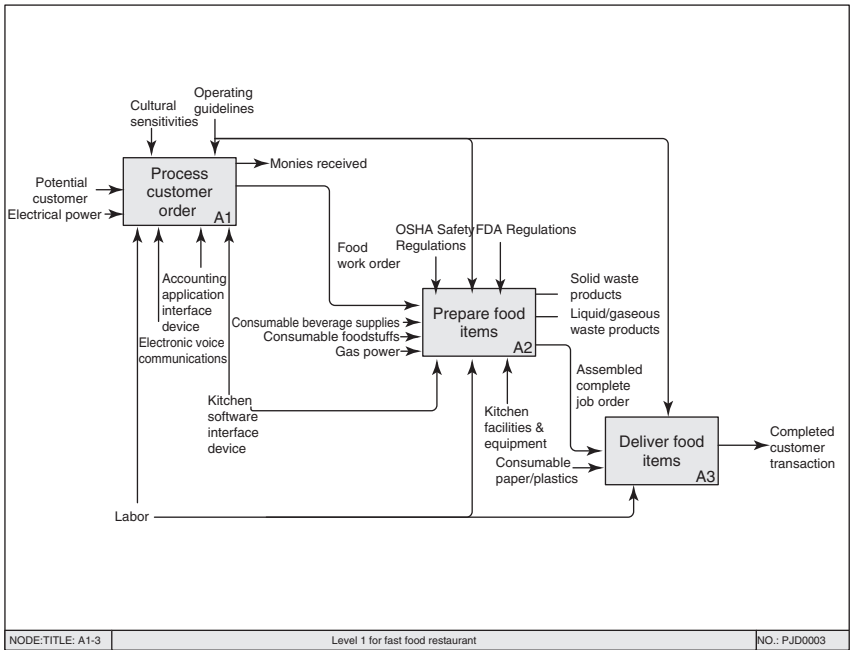

Figure 2.11 Level 1 model of a fast food restaurant.

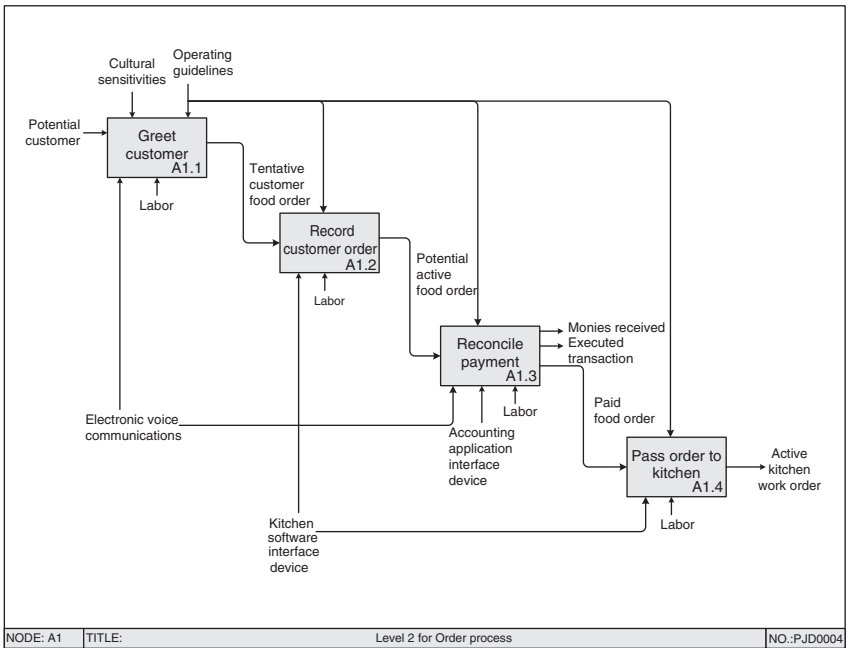

Figure 2.12 Level 2 model of the order process function for a fast food restaurant.

processes shown at Level 1. We leave these as an exercise for the reader. Notice how each level increases the amount of detail being presented concerning the system's operation, which has implications on the amount and quality of information needed from stakeholders in order to properly represent the system at these levels.

IDEF0 diagrams represent one modeling framework that has been implemented in a host of software applications. Since their introduction and with the explosive growth of interconnected systems, other tools strongly leveraging systems thinking have been developed. One such effort resulted in a modeling language based on the object-oriented analysis and design language Unified Markup Language (UML) called SysML.

The SysML (Systems Modeling Language) is a general-purpose modeling language for systems engineering applications that supports the specification, analysis, design, verification, and validation of a broad range of systems and systems-of-systems [19]. Introduced in 2003 by a group of partners interested in improving the precision and consistency of systems diagrams, the development effort split in 2005 into an open-source project (SysML: www.SysML.org) and a commercial software product (OMG SysML: www.sysmlforum.com) offered by the Object Management Group. Where UML is predominantly used for software development, SysML's charter extends into much broader classes of systems, many of which do not include software components.

As can be seen in Table 2.2, SysML contains a host of visualization tools that enable a user to represent a system in a number of ways depending on the need

TABLE 2.2 Open Source SysML Diagrams [19]

SysML Diagrams	Primary Purpose
Activity diagram	Show system behavior as control and data flows. Useful for functional analysis. Compare Extended Functional Flow Block diagrams (EFFBDs), already commonly used among systems engineers.
Block Definition diagram	Show system structure as components along with their properties, operations and relationships. Useful for system analysis and design.
Internal Block diagram	Show the internal structures of components, including their parts and connectors. Useful for system analysis and design.
Package diagram	Show how a model is organized into packages, views, and viewpoints. Useful for model management.
Parametric diagram	Show parametric constraints between structural elements. Useful for performance and quantitative analysis.
Requirement diagram	Show system requirements and their relationships with other elements. Useful for requirements engineering.
Sequence diagram	Show system behavior as interactions between system components. Useful for system analysis and design.
State Machine diagram	Show system behavior as sequences of states that a component or interaction experience in response to events. Useful for system design and simulation/code generation.
Use Case diagram	Show system functional requirements as transactions that are meaningful to system users. Useful for specifying functional requirements. (Note potential overlap with Requirement diagrams.)
Allocation tables	Show various kinds of allocations (e.g., requirement allocation, functional allocation, structural allocation). Useful for facilitating automated verification and validation (V&V) and gap analysis.

being addressed. Many of these tools can augment those introduced in later chapters to support the SDP activities.

2.7 MATHEMATICAL STRUCTURE

The structure of a system can also be described once a symbolic model of the system has been constructed using mathematical notation. Doing so requires us to focus not simply on the elements of a system but also on the relationships existing between elements. These relationships are the binding material of systems, and mathematics provides a means by which these relationships can be captured and analyzed. What is the nature of these relationships? What do they imply about the system itself? How do the relationships or the elements themselves change over

time? If allowed to continue in its current operational configuration, what might the state of the system be sometime in the future? Many of the modeling methods that follow in later chapters are chosen based on relationship characteristics.

An assumption at the heart of most mathematical system models is that the observable output of a system is a direct result of the input provided and the controls acting on the system. This relates input to output in a cause-and-effect manner that allows us to think of (a) the system's overall behavior within its boundary and (b) the functions and subfunctions supporting this behavior as mathematical functions.

In each of the IDEF0 model examples introduced in Figure 2.6, the bounding box containing the system function accepts input and transforms this input into output via some known or unknown transformation function that can potentially be expressed as a cause–effect relationship.

Consider the IDEF0 model for a system we label "collect intelligence" in Figure 2.13. Let (c) represent the input (cause) information to the system and let (e) be the output (effect) intelligence report [30]. Suppose that we impose a change in the input information (c), which we denote by Δc. This change in turn causes a change in the output e of the system, which we represent as Δe. The exact translation or transformation of this change, or *perturbation*, is accomplished by some transformation function which we denote as $g(e, c)$. The nature of $g(e, c)$ defines the mathematical structure of the system. Its action moves the system used to produce intelligence reports from one state $S(e, c)$ to another $S(e + \Delta e, c + \Delta c)$. This idea of a *system state* is a general concept that can be used to describe a system's condition, location, inherent health, financial position, and political position, among others.

Is $g(e, c)$ linear or nonlinear? We can determine this by examining the proportionality of the effect response for various changes in input. If Δe is proportional to Δc, then $g(e, c)$ is linear. If not, then $g(e, c)$ is either discontinuous or nonlinear. Determining the best mathematical form of $g(e, c)$ is left up to experimentation and data analysis using methods like linear regression, spline fitting, response surface modeling, and others. Once $g(e, c)$ is identified, is there a best level of output for given ranges of input that the system can be tuned to produce? Optimization techniques such as linear and nonlinear programming, equilibrium models, and related techniques could be used to answer this question and are introduced in Chapter 4.

Does the transformation performed by the system function stay consistent over time? If so, then $g(e, c)$ is likely *time invariant* and time is not explicitly modeled in the mathematical system representation. Otherwise, time should be included in the

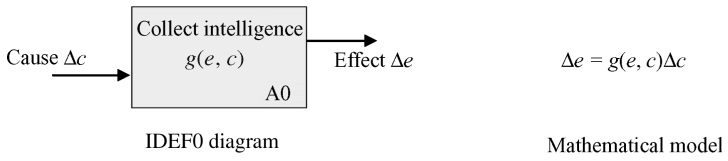

Figure 2.13 Two abstract system models: graphical and mathematical.

system function representation: $g(e, c, t)$, in some manner. If $g(e, c, t)$ is nonlinear, is the system *stable*, or do the effects increase without bound?

Is the relationship between Δc and Δe known with certainty? If not, then $g(e, c)$ should include probability or uncertainty elements. This is typically the case for systems that are best represented by simulations, queuing networks, decision analysis structures, risk models, forecasting methods, reliability models, and Markov processes, among others.

The role played by the function $g(e, c)$ in this abstract system representation is crucial. In mathematical terms, the function $g(e, c)$ is referred to as a *system kernel* [13]. It fundamentally describes the incremental change occurring between input and output to the system, which can occur in discrete steps:

$$g(e, c) \equiv \frac{\Delta e}{\Delta c} \tag{2.1}$$

or continuously:

$$g(e, c) \equiv \frac{de}{dc}. \tag{2.2}$$

The two expressions (2.1) and (2.2) are related by a limit expression as the incremental interval is made infinitesimally small:

$$\lim_{\Delta c \to 0} \frac{\Delta e}{\Delta c} = \frac{de}{dc} = g(e, c). \tag{2.3}$$

Figure 2.14 illustrates a slightly more complicated system structure to represent mathematically. The external feedback loop complicates the input to the system by adding a new input component that represents a manipulated portion of the system output. A common example of a structure of this kind can be envisioned by thinking about the creation of a political speech as an IDEF0 model's function, one element of a much larger political system. In this manner, we can construct a system boundary around the speech writing process from the perspective of a particular speech writer.

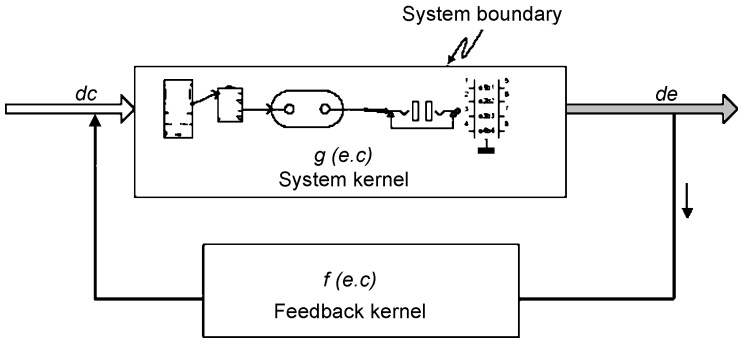

Figure 2.14 A system representation with external feedback.

Constantly changing information flows into this system from the environment, acting as input dc to the speech writing kernel, $g(e,c)$, which processes this input to create the output political message for a speech, de. However, the speech writer also wisely takes into account the public reaction to previous speeches when crafting a new message to be delivered. This consideration of public processing of previous output de creates a new system function called a *feedback kernel* which is part of an external system that we associate with the public. The output of this feedback kernel is *public* opinion relevant to the speech writer.

The public takes the previous output de and uses de as input to the feedback kernel $f(e,c)$. The output of the feedback kernel, $f(e,c)de$, is added to the normal environmental input dc so that the total (new) input to the system becomes $dc + f(e,c)de$ Thus we have

$$de = g(e,c)[dc + f(e,c)de] \tag{2.4}$$

Finally, by gathering common terms in Equation (2.4) and assuming that $dc \neq 0$, we see that the system kernel alters its form to a new form $g(e,c)_f$, where the subscript f simply designates that the feedback effects have been included in the system kernel:

$$\frac{de}{dc} = g(e,c)_f = \frac{g(e,c)}{1 - g(e,c)f(e,c)} \tag{2.5}$$

By thinking of the action $f(e,c)$ has on the previous system output de, and noting that the next output from the system kernel $g(e,c)_f$ is again acted on by the feedback kernel, we can easily get a sense that the feedback input is either being left alone ($f(e,c)de = 0$), amplified ($|f(e,c)de| > 1$), or dampened ($|f(e,c)de| < 1$) with each loop negotiation. This example begins to hint at some of the underlying system structure that can introduce system *complexity*, which arises from nonlinear dynamic changes, similar to those exhibited by this example when $g(e,c)$ is nonlinear. Complexity also arises when the system kernel $g(e,c)$ alters its structure in response to changes in input c. This ability of the system to adapt is again a characteristic of complex systems.

In an IDEF0 model representing sequenced functions, activities, or processes, feedback can also be represented in one of the two forms: control feedback and/or input feedback. Control feedback takes the output of a lower bounding box in the sequence order and connects it via an arrow to the control of an earlier bounding box in the sequence. A practical example of this can be seen in the organizational use of after-action reviews and similar activities that use system output to guide the system function. Similarly, input feedback takes the output of a lower bounding box in the sequence order and connects it via an arrow to the input of an earlier bounding box in the sequence. This situation is commonly encountered in rework situations—for example, in manufacturing or report writing.

Large systems are not necessarily complex systems. The presence of complicated and strong element interactions, nonlinear dynamic changes induced by system function kernels, and possibly self-organization behavior can impose complexity on

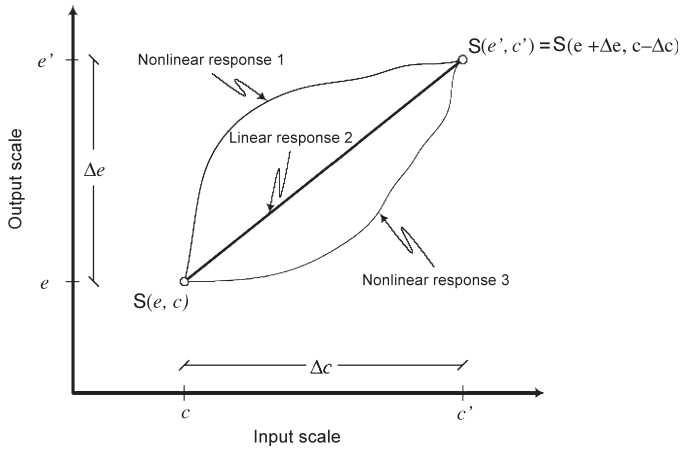

Figure 2.15 Three possible qualitative structures of a system kernel function.

a system; being large in size complicates an already challenging situation. Systems decision problems possessing some or all of these characteristics are recently being referred to as *wicked* problems [21]. Most often, the tool of choice to model and analyze a complex system is simulation.

Early in a system life cycle (see Chapter 3) when input and output data for a system are likely unavailable, it is still possible to gain a qualitative sense of the mathematical structure of a system using this same mathematical structure approach without knowing the exact notational form of either the system kernel $g(e, c)$ or existing feedback functions. Optionally, it may be possible to use a graphical approach in collaboration with the system owner or user(s) to extract a more general, but still useful, approximation of $g(e, c)$ by simply focusing on input and output to the system. Figure 2.15 illustrates three possible resulting function forms for $g(e, c)$ for the case when the system kernel is assumed to be continuous.

The topmost nonlinear curve 1 in Figure 2.15 describes a system kernel $g(e, c)$ that quickly drives the system state $S(e, c)$ to change for small changes in system input c. This shape is called *concave*. As the size of the imposed change on input Δc increases, the output response decreases. The bottommost nonlinear curve, 3, has just the opposite characterization. The system kernel $g(e, c)$ translates large perturbations on the system input into large output responses. This upward opening curve is called *convex*. The middle curve, 2, illustrates a proportional response that remains consistent throughout the range of input changes Δc. The particular shape of the system kernel estimated in this manner provides important clues as to what the mathematical model of this system function should be.

2.8 SPATIAL ARRANGEMENT

The natural state of affairs for systems is that they exist within and among other systems so that for any particular systems decision problem, a systems engineer is

inevitably faced with a composite system environment. Determining where and how the specific system under focus is located relative to the other systems it interacts with is a necessary task for setting the scope of the program. Some systems will have extensive interaction with the system under focus, which means that changes occurring in one of these interrelated systems have an impact on the behavior of the system under study. Systems of this kind have a high priority for being included within the scope of the program. As the degree of interaction between systems diminishes, systems falling into this category are more likely to be simply noted for the record and set aside.

One important note to consider is that systems often interact on an abstract level in addition to or in lieu of having linked system elements. Market effects, a consideration related to the financial environmental factor, are a good example of this. Competition for market share between—for example, Linux and Microsoft® WindowsXP operating systems—creates a strong interaction that must be considered when version updates and functionality revisions are being designed and developed. Likewise, two separate countries can be strongly linked politically because of shared vested interests even though no elements of their infrastructure are interconnected. Being sensitive to subtle interactions such as these and others helps identify and characterize lateral systems described in what follows. This is one of the uses of the environmental factors surrounding the four steps of the SDP shown in Figure 1.7 in Chapter 1.

A good visualization of the typical arrangement of systems interrelationships is useful for detecting the connections between various systems. Generally, systems either start out as, or evolve into being, part of other systems in one of three structural relationships: multilateral, lateral, and multilevel hierarchies. A multilevel hierarchical arrangement is one that is very familiar to large organizations because this is typically the manner in which they are structured to accomplish defined objectives. These arrangements are also referred to as *nested* systems. A more commonly encountered term is a "system of systems."

For a systems decision problem, recognizing and understanding the relationships between interacting systems, especially in a system-of-systems situation, is significant because these systems are likely to be at different stages in their individual life cycles. Mature systems tend to have a strong degree of presence in the composite system environment, an extensive network of connections, and significant resource requirements, and they are able to sustain a competitive situation for a long duration of time. In comparison, younger systems tend to have newer technologies, less presence in the composite system environment, less extensive interdependencies with existing systems, may have significant resource requirements, and their ability to sustain competition is less robust. Not recognizing this additional characteristic can result in unsatisfactory systems solutions being constructed. Moreover, notice that these observations, important as they are, say little about system efficiency, effectiveness, ability to leverage resources or arrange cooperative strategic alliances, and a host of other concerns that address the long-term survivability of systems. Each systems decision problem should be approached as being unique,

and it is up to the systems team to discern the relevant issues of concern from stakeholder and decision maker input to the SDP.

In a hierarchy, the system under study resides within a broader contextual arrangement called the *metasystem*. It in turn can have groupings of connected elements interacting to accomplish defined objectives. These are called *subsystems*. Within a hierarchical arrangement, like entities exist on the same level. Subsystems particular to these entities exist below them, and systems that they are a part of exist above them. Systems operating on the same level that share common elements are called *multilateral* systems. Systems on the same hierarchical level that are linked only by abstract or indirect effects (e.g., political influence, competition, targeted recruiting at the same population) are referred to as *lateral* systems.

An example of a multilateral arrangement from social network analysis [2, 22] is shown in Figure 2.16. Cindy is a person who exists as a friend to two separate groups of people that do not interact. The bridge relationship connecting the two social groups (systems) through the single individual (shared component) is multilateral systems structure. Changes that occur in one group have the potential for inducing changes in behavior in the other group through this shared component. The two multilateral friendship networks shown in Figure 2.16 would be considered to be in a lateral system arrangement if they did not share the element "Cindy" in common.

The positioning of a system within a hierarchy is relative to the system being examined, being determined in large part by the system purpose or how it functions in its relationship to other systems. It is this idea that motivates the construction of a functional hierarchy for systems decision problems.

Figure 2.17 presents two perspectives of the exact same group of hierarchical objects concerning the suspension system for an automobile. If the system we are concerned with is the vehicle's complete suspension system, we would conceptualize the system hierarchy consistent with Perspective 1: The system would reside

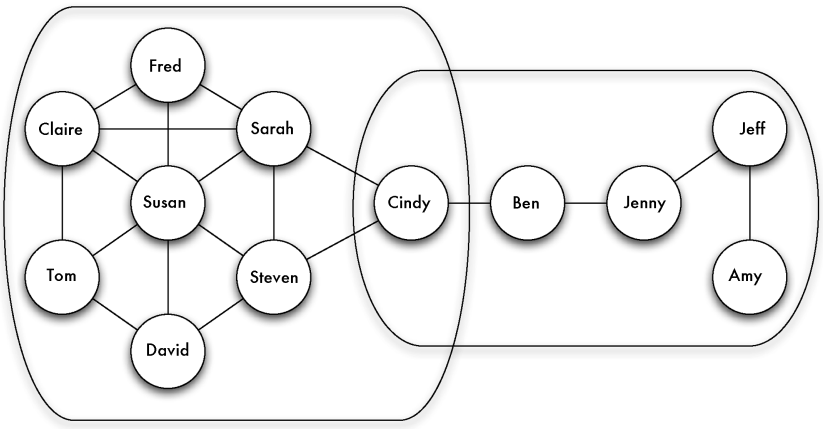

Figure 2.16 Multilateral friendship systems in a social network [1].

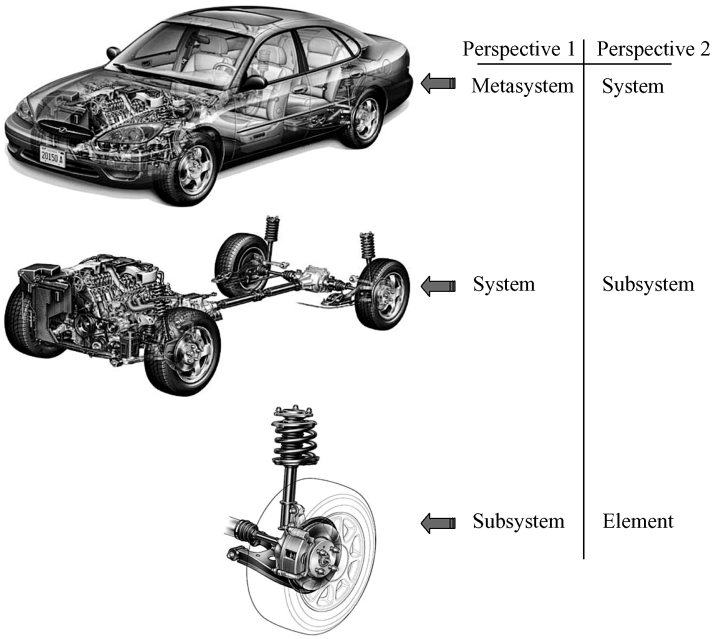

Figure 2.17 Three hierarchy levels of system spatial placement. (Courtesy of Kevin Hulsey Illustration Inc.).

within the metasystem of the entire automobile and have as one of its subsystems the left rear brake drum and shock absorbing subsystem. If we were instead focusing on the entire automobile as our system of concern, then Perspective 2 would be appropriate. The complete suspension system would be a subsystem and the left rear brake drum and shock absorbing object would be an element of this subsystem. One possible metasystem for Perspective 2 could be the rental car system that uses this automobile as part of its available fleet.

In hierarchical systems, the observable system behavior can easily change, depending on the system being examined. At higher levels in a hierarchy a more abstract, encompassing view of the whole system emerges without attention to the details of the elements or parts. At lower levels, where subsystems and individual elements are evident, a multitude of interacting parts can typically be observed but without understanding how they are organized to form a whole [23].

In a pure multilateral arrangement in which the only connections between systems are shared elements, the relationship between these systems is essentially nonhierarchical. For situations such as this, it is then valuable to conceptualize the arrangement of systems as a *composite multilateral* arrangement [24]. Subsystems and elements can then have the same meaning as in a typical hierarchical structure. The environment shaped by the interactions emanating from the composite multilateral systems takes the place of a metasystem. This is a particularly useful construct for systems decision problems involving strategy and policy, where the

dynamics of influence, power, or counteraction are key to shaping the decisions being made. Recent examples of such an arrangement arose when applying systems thinking to U.S. border security policy [25], metropolitan disaster planning [24], and counter-insurgency strategy [26].

2.9 EVOLUTION

While we may observe systems within a limited period of time, and subsequently gather data that we use to design, improve, or change systems, we must always bear in mind the fact that the system does not sit still and wait for us to finish what we are doing. A system is somewhat like an object in a pond. An extremely hard object like a glass ball would change little over time in this environment, yet the environment may radically change around it. Conversely, while the environment in a pond surrounding a saltine cracker may not change much, the internal structure of the cracker would indeed change over time. And so it is with systems.

Systems engineers concern themselves with how system elements interact both within and external to system boundaries at the time they are observing the system. They must also be aware of how these interactions and conditions affecting them might evolve in the time that passes between different team actions. For example, performance data for individual securities on the New York Stock Exchange would have to be continually updated in a systems decision problem concerning portfolio structuring. If this is not done, a potential investment strategy that would work at the time of data collection might not a week, month, or 6 months later. The sensitivity of these solutions to time is very large. The system dynamics are rapid, they have the potential for wide variation in performance, and they are filled with uncertainty. Any system representation and subsequent program planning should take these characteristics into consideration.

Chapter 4 introduces several techniques for modeling and analyzing systems and their effects when evolution is a consideration, whether the changes of interest are considered continuous or discrete. A continuous system is one whose inputs can be continuously varied by arbitrarily small changes, and the system responds to these continuous inputs with an output that is continuously variable. A discrete systems is one whose inputs change in discrete amounts (e.g., steps in time, fixed increases in volume, number of people) that generally have a minimum size below which it does not make sense to reduce. Thus, the output tends to respond in a stepwise fashion as well [13].

2.10 SUMMARY

Systems thinking distinguishes systems engineering from other engineering fields. The holistic viewpoint that embodies this perspective makes the systems engineer a valued and unique contributor to interdisciplinary team efforts. Adopting this world view enables systems to be identified, classified, and represented in such a way as to lead to a deeper understanding of the system's inner workings and interconnected

relationships that contribute to system-level behavior. The end result of applying this thinking using the SDP directly contributes to a more comprehensive, more robust, and richer array of possible solutions generated for a systems decision maker to consider.

Systems thinking also affords an ability to establish a proper system boundary, scope, and orientation for a systems decision problem regardless of whether the system is open or closed to its environment. Knowing this information keeps an interdisciplinary systems team focused on the problem at hand and provides an understanding as to where their most significant contributions can be made.

Structure defines the interconnected relationships between system elements and other entities. This structure can be represented in a variety of ways, some mathematical as in input–output functions and systems kernels, and others graphical, as in IDEF0 models. A good start toward understanding this structure can be gained using qualitative methods during stakeholder analysis. The underlying structure of a system can be complicated when internal or external feedback is involved because the natural inputs seen by a system are being preconditioned, possibly without the system's internal elements being aware of it. Moreover, "downstream" output could be intentionally or unintentionally exerting nontrivial controls over early functions in a sequenced process.

Systems naturally contain an internal hierarchical arrangement of elements and processes. Systems also exhibit hierarchical arrangements and ordering with respect to their relationships with other systems. Identifying the structure and form of these hierarchical arrangements during the Problem Definition phase of the SDP leads directly to the critical and necessary understanding that supports the decision gate leading to effective Solution Design, the next phase of the SDP.

2.11 EXERCISES

2.1. For each of the following systems, characterize them as a white box, gray box, or black box system from a particular perspective that you identify. Briefly explain your reasoning and what could be done to increase the level of internal understanding of the system, if appropriate.

(a) Political system in effect governing the activities of the office of Mayor in Providence, Rhode Island.

(b) A magic act you see performed at Lincoln Center in New York City.

(c) A slot machine in operation in Reno, Nevada.

(d) The Dave Matthews Band.

(e) A crime syndicate operating in Newark, New Jersey.

(f) A vacation travel agency.

(g) A Blackberry® device.

(h) Gasoline consumption in the United States and global warming.

(i) A George Foreman® "Champ" grill.

(j) The admission system to college.

2.2. Identify possible sources of internal and external feedback that could or does exist for the systems in the previous question. Identify any possible ways of quantifying this feedback where it exists.

2.3. Using Figure 2.9 and Figure 2.8 as guides, construct IDEF0 Level 1 models for the Solution Design, Decision Making, and Solution Implementation phases of the SDP. Since Figure 2.9 was created using the first edition SDP, change the Level 2 IDEF diagram to accommodate the new SDP elements of the Problem Definition phase shown in Figure 1.7.

2.4. Referencing Figure 2.11, create IDEF0 Level 2 models for the fast food subfunctions "prepare food items" and "deliver food items."

2.5. Create IDEF0 Level 0 and Level 1 models for the following systems:

(a) Harley-Davidson® Sportster motorcycle.

(b) The U.S. presidential election process.

(c) Scuba gear.

2.6. Create a Level 0 model that represents the political speech writing process introduced as an example in Section 2.7. Be sure to include any intended and unintended consequences of the process as output of the model. What controls are imposed on this process?

2.7. What classification would you assign to the following systems:

(a) Maslow's hierarchy of needs (Figure 2.18)

(b) Algebra

(c) The Ten Commandments, Torah, and/or Quran

(d) Marriage

(e) Motor vehicle operating laws

(f) Your study group for a class you are taking

(g) A human heart

(h) A cell phone

(i) An Ipod

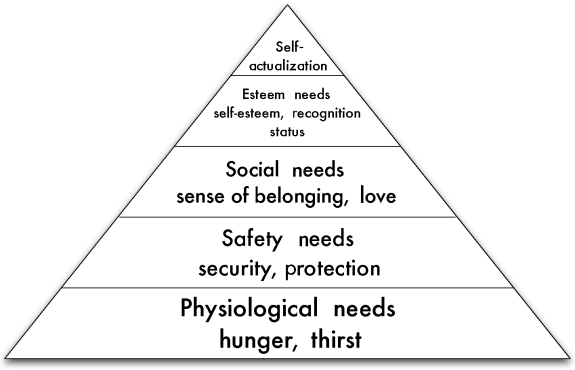

Figure 2.18 Maslow's hierarchy of needs.

2.8. Create an IDEF0 Level 0 model for a car purchasing decision as a system from your point of view.

2.9. Why is the Linnaean taxonomy a hierarchy? (www.palaeos.com)

2.10. What type of structure does the Department of Defense's Global Information Grid (GIG) have? Briefly explain. Cite credible resources on the Internet that you use to answer this question.

2.11. In recent times, as organizations have become more complex, corporate leaders have examined the idea of a "flat organization" to explore whether it would operate more effectively and efficiently. In particular, recognizing that the environment around organizations is constantly evolving, one wonders whether a flat organization would outperform a traditional hierarchy in terms of being able to successfully adapt to changes in its environment. Use credible web sources to define a flat organization. Select a major organization that you are familiar with and develop a supported position as to whether it would be better off in some capacity if it transitioned to a flat organization.

2.12. Rensis Likert has conducted extensive research on a nonbureaucratic organization design referred to as System 4 (participative versus democratic). Is System 4 a Multilateral or multilevel arrangement? Explain.

2.13. What type of structure does the U.S. Navy's Trident weapon system have? If you isolate the warhead as a system of concern and were to initiate a systems engineering study on it, what other systems would interact with it?

2.14. The U.S. Department of Homeland Security announced a call for proposals in 2005 for a multi-billion-dollar project called Secure Border Initiative (SBI*net*) to test the ability of technology to control U.S. borders after a succession of failures. What type of structure are they envisioning for the SBI*net*? Suppose that you were going to conduct a systems engineering study that focused on the U.S. border as a system.

(a) Apply systems thinking to this problem by using the environment factors of the SDP to construct a systems hierarchy that shows the major lateral and multilateral systems, subsystems, and metasystems that define the U.S. border.

(b) Would a policy of limiting immigration to the United States be a system-level or symptom-level system solution? Explain.

(c) Which of the systems that you specified in part (a) would you consider to evolve over time?

(d) For each of the evolving systems you identified in part (c), list one symptom-level and one system-level phenomenon that you think provide evidence that the system is evolving.

2.15. Using a satellite imagery mapping service such as Google Maps or Google Earth, locate the U.S. Customs Port of Entry at the north side of Heroica Nogales. Now, locate the U.S. Customs facility where U.S. Highway 91 becomes Canadian Highway 55 on the northern border of Vermont.

(a) Thinking of these two border crossing locations as systems, how do the systems represented by the environmental factors of the SDP differ between the two locations? For example, does their importance change, depending on which location you are working with?

(b) At which location would a pure technological solution like remote cameras work better?

(c) Compare the stakeholders at both locations. Why might they be different? Which are in common?

(d) For the stakeholders that you identified as being in common at both locations, would you expect their vested interest (stated position) on border security to be the same for both? What does this imply for a single immigration policy for the United States?

(e) For each of the stakeholders you identified in part (c), use the internet to find credible source documentation that enables you to identify their vested interests with regards to the border security issue. Were their any surprises?

2.16. From the viewpoint of a winery owner, the process of making red wine can be conceptualized as a system comprised of three major functions: preparation, fermentation, and aging.

(a) Construct an IDEF0 Level 0 model for a California winery currently in operation that you identify from the Internet. Assume the viewpoint of the winery owner.

(b) Using the three system functions listed, construct an IDEF0 Level 1 model for the winery that properly illustrates the sequence of these functions along with their inputs, outputs, mechanisms, and controls.

(c) Construct an IDEF0 Level 2 model for the "fermentation" function, basing this model's detail on actual information for the winery you choose that is provided by credible Internet references.

(d) Estimate the mathematical representation that could be used for the fermentation function kernel.

(e) Would the Level 0 model change if the viewpoint adopted was that of a distributor or retail wine store operator? Briefly explain.

2.17. Figure 2.19 was created in support of a systems engineering study focusing on information flow of communications on the modern battlefield [27].

(a) There appears to be a natural hierarchy present. How would you represent this in a diagram?

(b) What systems thinking ideas can you apply to this illustration? What structure is attempting to illustrate? Can you identify any elements of complexity? What would a systems component be in this illustration? Does the system behavior change at different levels being observed?

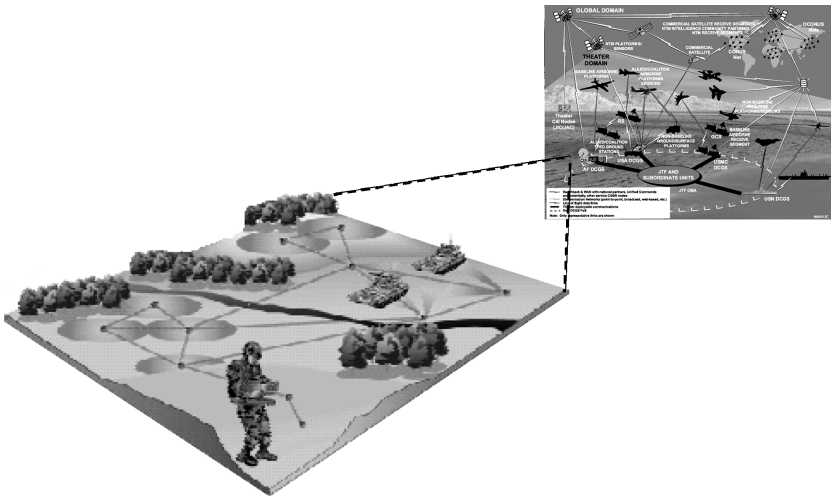

Figure 2.19 An illustration of information flow.

REFERENCES

1. Pidd, M. *Tools for Thinking*. Chichester, West Sussex, England: John Wiley & Sons, 2003.
2. Wasserman, S, Faust, K, Iacobucci, D, Granovetter, M. *Social Network Analysis*: *Methods and Applications*. Cambridge, England: Cambridge University Press; 1994.
3. Haines, SG. *The Manager's Pocket Guide to Systems Thinking and Learning*. Amherst, MA: Centre for Strategic Management, HRD Press, 1998.
4. Okes, D. *Root Cause Analysis*. Milwaukee, WI: ASQ Quality Press, 2009.
5. Jamshidi, M. *System of Systems Engineering*. Hoboken, NJ: John Wiley & Sons, 2009.
6. Hybertson, DW. *Model-Oriented Systems Engineering Science*. Boca Raton, FL: CRC Press, 2009.
7. Aslaksen, EW. *Designing Complex Systems*. Boca Raton, FL: CRC Press, 2009.
8. Markowitz, HM. Portfolio selection. *Journal of Finance*, 1952;7(1):77–91.
9. McDermott, I. *The Art of Systems Thinking*. Hammersmith, London, England: Thorsons Publishing, 1997.
10. Skyttner, L. *General Systems Theory*: *Ideas and Applications*. Singapore: World Scientific Publishing Co., 2001.
11. Bullock, A. *The Secret Sales Pitch*. San Jose, CA: Norwich Publishers, 2004.
12. Abbott, EA. *Flatland*: *A Romance in Many Dimensions*. New York: Dover Publications, 1952.
13. Sandquist, GM. *Introduction to System Science*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
14. Mayer, RJ, Painter, MK, deWitte, PS. *IDEF Family of Methods for Concurrent Engineering and Business Re-engineering Applications*. College Station, TX: Knowledge-Based Systems, 1992.

15. Colquhoun, GJ, Baines, RW, Crossley, R. A state of the art review of IDEF0. *Journal of Computer Integrated Manufacturing*, 1993;6(4):252–264.
16. George, ML, Rowlands, D, Price, M, Maxey, J. *Lean Six Sigma Pocket Toolbook*. New York: McGraw-Hill, 2005.
17. Fox, N. Case Study: Process definitions–Process measuring; how does a small foundry get their ROI off ISO/TS 16949:2002? *AFS Transactions 2005*. Schaumburg, IL: American Foundry Society, 2005.
18. Straker, D. *A Toolbook for Quality Improvement and Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
19. Open source SysML project. Available at http://www.SysML.org. Accessed 2010 April 13.
20. Sandquist, GM. *Introduction to Systems Science*. Englewood Cliffs, NJ: Prentice-Hall; 1995.
21. Yeh, RT. System development as a wicked problem. *International Journal of Software Engineering and Knowledge*, 1991;1(2):117–130.
22. Krackhardt, D, Krebs, V. *Social network analysis*. Semantic Studios Publication; 2002. Available at http://semanticstudios. com/publications/semantics/000006.php]. Accessed January 6, 2006.
23. Heylighen, F. 1998. Basic concepts of the systems approach. Principia Cybernetica Web, (Principia Cybernetica, Brussels). Available at http://pespmc1.vub.ac.be/SYSAPPR. html]. Accessed January 7, 2006 .
24. Driscoll, PJ, Goerger, N. Stochastic system modeling of infrastructure resiliency. ORCEN Research Report DSE-R-0628. West Point, NY: Department of Systems Engineering, U.S. Military Academy, 2006.
25. Driscoll, PJ. Modeling system interaction via linear influence dynamics. ORCEN Research Report DSE-R-0629. West Point, New York: Department of Systems Engineering, U.S. Military Academy, 2006.
26. Driscoll, PJ, Goerger, N. Shaping counter-insurgency strategy via dynamic modeling. ORCEN Research Report DSE-R-0720. West Point, NY: Department of Systems Engineering, U.S. Military Academy, 2006.
27. *Implementation of Network Centric Warfare*. Office of Force Transformation, Secretary of Defense, Washington, DC, 2003.
28. Jackson, MC. *Systems Thinking*: *Creative Holism for Managers*. West Sussex, England: John Wiley & Sons, 2003.
29. Pidd, M. *Systems Modelling*: *Theory and Practice*. Chichester, England: John Wiley & Sons, 2004.
30. Driscoll, PJ, Tortorella, M, Pohl, E. Information quality in network centric operations. ORCEN Research Report. West Point, New York: Department of Systems Engineering. 2005.
31. *Rethinking Risk Management in Financial Services*. World Economic Forum Report 110310, New York, April 2010.