Chapter 4 – An Introduction to the Process of Understanding Systems

Abstract

Recalling that all systems are processes at some scale, the inverse is true, that all processes are systems. Understanding complex systems will require a system of inquiry, a systematic process. In this chapter, we provide an overview of the whole system understanding process, *presented as a system*. As outlined in the Introduction this process is comprised of seven major sub-processes, each of which can be further deconstructed into sub-sub-processes (an example is given in Figure 4.1 below). After introducing the whole system and developing the initial deconstruction the chapter will provide a general description and brief explanation of each of these major sub-processes. Each will be more fully explained, with examples of how the processes work, in Parts 2, 3, and 4.

4.1 From General Systems Principles and Theories to Actual Systems Knowledge

Let us utilize the principles of systems science discussed in Chapter 1 to design a process of analysis and synthesis (maintaining a holistic knowledge) that will allow us to understand complex systems as described in the Preface and Introduction. This system of analysis and synthesis is the product of using the process that is to be summarized in this chapter and elaborated in the balance of the book. That is, reflexively, this process is a product of itself. To explain: The author has developed this process over years of studying real-world systems using elements from this process without having formalized their relations in the way developed in this book. Only after completion of the Principles book (Mobus & Kalton, 2015) did the author realize that organized appropriately these various methods, i.e., deep analysis, knowledgebase construction, and generating/testing models from the knowledgebase, constituted a whole process or a system in its own.

This experience is a microcosmic version of the story of humanity evolving the formal systems we have for gaining increasingly veridical knowledge about parts of the world – the sciences and maths. We begin with that story because it is instructive with respect to how we can gain systems knowledge by the process described here. But unlike knowledge gained within a single disciplinary silo, systems knowledge is transdisciplinary allowing us to tackle the most complex kinds of systems we want and need to understand.

4.1.1 From Observing to Characterizing to Modeling

As described in the Introduction, we human beings have been most successful in our ability to interact with all of the various environments on the planet by virtue of our ability to more deeply understand the systems with which we interact in those environments. We can anticipate the near future given our knowledge of how things work and prediction of how they will behave. For most of human prehistory and even into the Bronze Age, humans could use their natural, intuitive, capabilities to understand their world based on the language of thought, which, as asserted in chapters 1 and 3, is actually systemese. As long as humans could observe the systemic aspects of the entities and processes in their environment, they could infer regularity of behaviors that served for anticipating the future. It also accounts for early humans who left Africa to adapt fairly quickly to very different environments. Systemness is the same everywhere even though the systems may seem very different superficially.

But the very capabilities that served so well as intuitive thinking produced a world of increasing complexity in which mere intuitive thinking would not lead to deep (enough) understanding, at least for individuals. Specialization, already incipient in tribal life (see discussion of the origins of economic culture in Chapter 8), became increasingly necessary because the breadth and depth capacity of the individual brain is limited (as discussed in the Introduction, polymaths are quite rare).

Science and math/logic were invented (or discovered through trial and error at first) to begin formalizing the process of gaining much deeper understanding of the world. Formalization consists primarily of developing the general pattern recognition of mathematics and logic so that a set of patterns could be applied to multiple different specific domains. Math and logic were elevated to methods of finding and exploiting patterns according to generalized sets of rules that did not depend on any particular substrate system. The scientific method (and the science process) is formalizations for acquiring direct knowledge within substrate domains through the application of mathematics and logic to generate theories and hypotheses, and hypothesis verification or falsification through empirical testing. These formalized processes have worked amazingly well despite the fact that many of the domain specific methodologies were developed in a somewhat ad hoc or independent (from one another) approach. It wasn't until philosophers of science, like Karl Popper[1], explored the epistemological aspects of modern science in general that an overview of the common approaches to all sciences began to emerge.

The sciences and engineering processes have advanced significantly with the maturing of formal methods. One consequence has been a significant jump in the complexity of the subject domains. Biology has gone from a descriptive discipline (naturalist studies) to one dependent on mathematical modeling (population studies, protein folding, and many others). The term

[1] See the Wikipedia article: https://en.wikipedia.org/wiki/Karl_Popper for background on Popper's influence on the epistemology of science. Accessed 12/25/2016.

'systems biology' is used to describe the new mathematically oriented approaches to getting deep understanding in the domain. The same phenomenon is taking place in the so-called hard sciences as well as the social sciences. Models are being developed to gain deep understanding of many phenomena. But there is more to systems <subject> than just mathematical modeling.

There is a growing realization that there are many patterns of organization and behaviors that transcend the boundaries of disciplines. For example, the infamous logistic (S-shaped) curve associated with population growth is found to be useful in many different domains where some value is increasing, at first exponentially positive, and then after some time, turns exponentially negative[2]. Many 'growth' processes can be modeled using this formulation with appropriate choices for the parameters.

Early pioneers of general systems theory recognized that there were general principles about nature that appeared to operate in all explored domains, which they termed 'isomorphies' (Troncale, 1978) and realized that all of these phenomena display the properties of systemness, hence, their start at codifying aspects of systemness under the rubric of General Systems Theory (GST, von Bertalanffy, 1969). At present, there are still many 'versions' of what is purported to be a GST; many researchers have rekindled the vision of the pioneers and seek some consensus on a general set of principles that would constitute a science of systems (systemology, c.f. Rousseau, et. al., 2016). The author along with co-author Michael Kalton attempted to collect the range of identified concepts of systemness and organize them into a principles framework in Mobus & Kalton (2015).

Though many would argue that we are still in the early stages of developing a consensus of what the principles are and how they should be organized, there has been significant progress made in that direction. In this author's view, we have enough background now that allows us to begin mapping the understood principles of a GST onto real systems (Checkland, 1999, page 9). We present here a methodological framework for accomplishing this task.

4.1.2 All Sustainable Processes are Systems

The process of understanding a system deeply (from principle #11) is shown to be a system! This follows from principle #2 summarized in Chapter 1. Processes are defined as sustained only if they are in some way bounded and all sub-processes contribute to the purpose of the whole in a balanced way. They take in inputs, process them, and export outputs (products), thus they are systems by that definition (Chapter 3). In fact, as argued previously system ↔ process; all systems are also processes! Thus, we talk about the system for understanding complex systems, the process of analyzing systems (and their environments), capturing the structured knowledge, and using that knowledge for multiple purposes in exploiting the knowledge.

[2] The classic population model is given by: $P(t) = \frac{KP_0e^{rt}}{K+P_0(e^{rt})-1}$, P_0 the initial population size, K an upper limit on P, r is the rate constant, and t is the time.

Below we present the transparent box system for understanding as a model (Figure 4.1). We will briefly analyze the whole system in this chapter in more detail as compared to Chapter 1 in order to provide an overview of the whole. Parts 2 and 3 will be elaborations of the components presented below and examples from multiple disciplines to show how they apply.

4.1.3 A Generalized Process Applied to Any Domain

What follows is effectively a generalized model for the process of understanding complex systems. This is possible because the process is based on the general principles of systemness and those apply to any kind of system from any domain of science (physical and social). This will be demonstrated in chapters 6 and 8 specifically, but also with examples sprinkled throughout the other chapters.

There is something philosophically satisfying about the idea that a conceptual system can be devised, and then implemented in a physical embodiment, that can then be used to expose the systemness of real objects in the world. Closure under systemness? Of course, the implementation must make provisions for particularizations, such as terminologies associated with the lexicon developed in Chapter 3, the language of system. The generality of the language (due to the generality of the ontology in Chapter 2) and the methods for particularization make the process extremely powerful for explicating specific kinds of systems. But at an even deeper level, the generality and capacity for particularization makes the process a viable tool for cross-disciplinary communications. It is hoped that this will permit the re-integration of the sciences in general. This is the ideal goal of transdisciplinarity – the ability to work seamlessly across what now seem to be isolated scientific (perhaps even humanities) silos of knowledge. Indeed, in Chapter 7 we will introduce the possibility of a universal knowledgebase that could integrate knowledge from across all of the sciences based on the proposition, claimed in Chapter 2, that the Universe is a system, in which all else are subsystems. Who knows, we might actually be able to eventually answer Lewis Carroll's quintessential question: "Why is a raven like a writing desk?"[3] For starters, our solution is that they are both systems!

4.2 The *System* for Understanding Complex Systems

The system for understanding is based on the archetype model of a complex, adaptive, and evolvable system (CAES) to be explicated in Chapter 9, as a transparent-box, consists of seven operational level subsystems including a governing process. Figure 4.1 provides an abstract overview of this system. Below we will explain each of these components and its contributions to understanding.

[3] In *Alice's Adventures in Wonderland*, the Mad Hatter's question to Alice. See the Wikipedia article: https://en.wikipedia.org/wiki/Alice%27s_Adventures_in_Wonderland. Accessed 12/28/2016

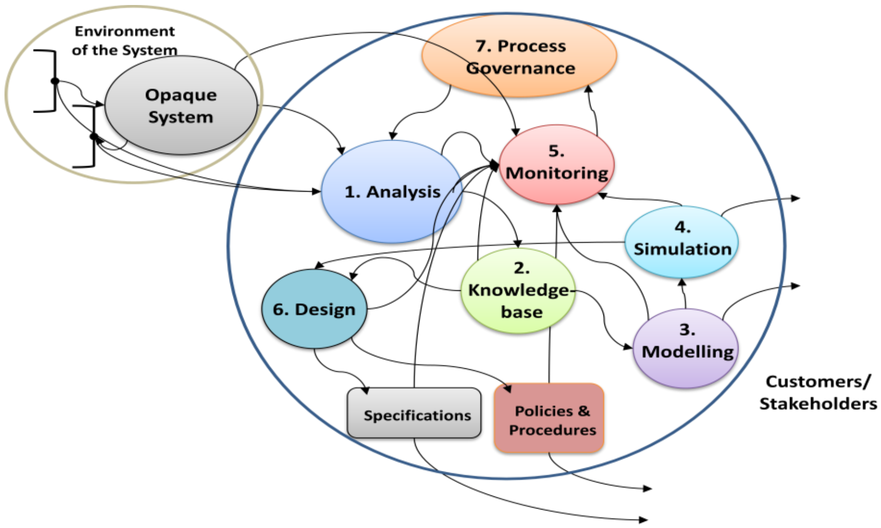

Fig. 4.1. The system for coming to deep understanding of other systems is comprised of the seven sub-processes (described in the text) including a governing process. The latter will be developed further in Chapter 11. The opaque concrete system, whether existent or to be designed, along with elements of its environment, is the input to the process. Various products are produced and distributed to customers and stakeholders. The arrows showing flows are of messages that convey information so long as there is a disequilibrium between the knowledgebase (and its outputs) and the analyzed system. Note that the 'specifications' and 'policies & procedures' are product outputs from Design. Specifications pertain to engineered systems, while policies/procedures pertain to other governance subsystems.

Much of what will be presented in this chapter will easily be seen to relate to human designed artifactual systems, which will be more fully developed in Part 4. However, it is important to point out that the process we are developing is as useful in coming to understand complex natural systems. That is, natural systems, such as complex ecologies, even whole planets, involving multiple disciplinary sciences can benefit from the system of understanding. In this situation the process works to provide a means for integrating knowledge domains which is often necessary in order to grasp the nature of emergent phenomena. Each of the scientific disciplines are applied to their domain piece of the system but the use of systems science as meta-science in the framework, so far disclosed, acts to integrate and synthesize whole systems structures/functions.

This is actually very similar to how the process works for systems engineering where the job of the systems engineers is to provide an integrative synthesis of subsystems developed by disciplinary engineers. The same process applies to both scientific and engineering understanding of the complexity with which they are faced in discovery or invention.

4.2.1 The General Flow Through

As described in the introduction, the process we describe here differs from traditional approaches in that information flows through the process generally starting with more rigorous systems analysis and knowledge capture to provide the basis for constructing various kinds of models and simulations (see Box 4.1, "Getting the Horse in Front of the Cart" below). This general flow is actually reverse of how most understanding processes, e.g. the scientific method or classical systems analysis, have been conducted in which some initial understanding (i.e. opaque-box analysis such as system identification) leads to model constructions followed by testing (i.e. solving or simulating and predicting then comparing to the real system behavior). In the more traditional approach, the model of a system is a form of hypothesis about how a system works. The model captures what are hoped to be relevant variables and relations, when solved or simulated the results should match what the real system does. If it doesn't the conclusion is that the hypothesis is wrong (disconfirmed) and a new version of the model is constructed and tested. Hypothesis are informed guesses as to how something works or what it will do in the future given certain inputs (environmental conditions). But what we hope to show is that this approach, while historically necessary and scientifically valid, is highly inefficient and becoming increasingly unnecessary since better deconstruction tools are coming on the scene almost daily. Along with the development of incredibly sensitive sensor technology and "Big Data" analytical methods it is becoming possible to design deconstruction methods for every scale of system we find interesting from planets to economies to organizations to people to ecological systems…well, to just about everything[4].

The emphasis in this new process is on deep analysis first and model building second. Modeling can still provide useful feedback to the analysis process (through the governance subsystem in Figure 4.1). There can still be iterations when discrepancies occur. No analytic technique will always be foolproof so some confirmation process is still required. But the number of iterations should be greatly minimized when the original analysis is done well and the knowledgebase captured is more complete than an opaque-box analysis can provide.

The principle advantage of putting more emphasis on deep analysis and knowledgebase production, over modeling first, is a gain in efficiency of the process and a tremendous saving of time in coming to understand what is going on. The model-and-see approach is logically valid (in the same sense that inductive reasoning is valid[5]), but generally takes much more time to work through. And there will always be a lingering doubt about the validity of the model – have we thought of every possible set of inputs? The deep analysis first (and foremost) approach should save considerable time and increase confidence in model outputs (the models are based on real knowledge and not guessed).

It will, most likely, be difficult for many scientists to adopt this viewpoint since there is a long tradition of exploring phenomena by proposing models (hypotheses) first, or as soon as some opaque-box analysis has produced seemingly reasonable results. The traditional approach

[4] For example, plans have been put forth to 'instrument' whole cities, e.g. place remote temperature, activity, and other kinds of sensors in key locations to collect data about conditions in those locations throughout the day as part of the city planning process.

[5] See, for example, the Wikipedia article: https://en.wikipedia.org/wiki/Inductive_reasoning for background. Accessed 1/13/2018.

has generally worked in the long run, and this new approach might, at first, appear to be too much work at the front end, so why break tradition. We human beings can adopt fairly conservative positions at times.

On the other hand, there is already some growing evidence that attitudes toward deeper analysis are already evolving. In the social sciences, in particular, but also in the life sciences and medicine where complexities are substantial, there is movement away from reliance on ad hoc models (especially from the opaque-box perspective) toward a greater effort to understand what is going on 'under the hood.' For example, in the field of economics much more attention is being focused on actual microeconomic mechanisms in attempts to understand 'real' macroeconomic processes (see Chapter 8). Coupled with the tremendous advances in computing power of the last decade (including the potentials of quantum computing in the not-too-distant future), along with the advent of 'Big Data' analytics, machine learning, and other artificial intelligence advances, our technical capacity to do deep analysis is not lost on many scientists. It is hoped that the methodologies presented in this book will provide a framework by which these advancing tools can be used in a principled fashion.

4.2.2 Component 1 – Deep Analysis

Here 'analysis' means transparent-box methods. This is what we will refer to as 'Deep Systems Analysis' or DSA. The system is deconstructed (decomposed) carefully to reveal the major inner workings, the internal components and their interactions, with the capture of this knowledge in a structured database (the knowledgebase – Component 2 below). Analytical methods revolve around instrumentation of systems so as to capture real data from 'inside' the system, from each of its subsystems, rather than from an opaque-box analysis where only the inputs and outputs to/from a system are monitored.

4.2.2.1 State of the *Art*

The word *Art* is emphasized when referring to what has passed as analysis for good reason. Current approaches to systems analysis derive from projects in logistics and weapons design during WWII when the concept of systems started to take hold. With the advent of computerized procedures used in business and government, as work systems got more complex, the term was applied to finding out what kind of information system would be needed for operations. The systems approach was still quite a new way of thinking in the late 1950s and 1960s as information technology was rapidly invading commerce and government. So, the notion of systems analysis was still very naïve. Most engineers and systems analysts (in job title) considered the system as either a "product" (i.e. the airplane) or the IT operation in support of the business (the management information system – MIS). They did not recognize that **THE** system was the whole operation of the business or agency with the MIS component being just that, a component of the larger work process. Their focus tended to be on the subsystem. An airplane is a product, but the whole airplane-making business along with its market environment is **the** system that needs to be deeply understood in order to make the best product possible. The

MIS is only a support system for the management decision processes that are collectively **the** governance subsystem. Not really understanding this distinction has been the source of many failures in engineering and information systems projects over the last many decades since the ideas developed in WWII were (partially) transferred to the business world.

It is an unfortunate fact that systems analysis has gotten a bad reputation in various contexts. We think the reason is actually pretty clear. Most systems analysis procedures are not actually analyzing the whole system in terms of gaining deep understanding of what it does or is supposed to do. Recall that deep understanding means the application of all of the principle ideas in systems science to finding out what is going on inside and out. Most often, for cost, or time, constraints the kind of deep understanding promoted here is foregone under the assumptions that the analysts involved already know what needs to be done and there is a leap to the specification writing phase before anyone is quite sure they have gathered all of the details needed to succeed. Most often the analysts have not got a clear idea of what to do and end up making assumptions that turn out to be wrong.

For example, in critiquing systems analysis of management decision systems where the objective has been defined according to systems engineering practices (‘hard’ systems) Checkland (1999) observed the problem with jumping to the conclusion that an existing management decision system (say, paper based) would be a sufficient model/specification for the new, improved system. Referring to the analysis method used to develop a state-level information system, he gets right to the heart of the problem:

> All of this, of course, eliminates from the study many of the most interesting questions, especially those concerning the purposes served by the existing flows of information and the desirability of others. Ruled out of the study from the start was any consideration of the *meaning* of the information flows in relation to decision-making at the state level. (Checkland, 1999) [italics in the original]

For Checkland (1999, pages 143-144) the main problem with using systems analysis as developed for engineering and operations research problem-solving was that once an objective has been identified the engineer assumes that her job is to simply find the best mechanism (meaning, perhaps, the most cost effective) for accomplishing that objective. In working with complex social organizations involving management decision processes, the objectives are not necessarily that clear, what an engineer would say are ‘under-specified’. Moreover, such systems have a tendency to change even while the engineering phase is underway. Checkland identified five arguments for why the hard engineering approach to systems analysis would not work. And he thus concluded that another approach, equally systemic, which he called ‘soft’ systems, would be needed.

Thus, there are today various versions of systems analysis that range from traditional engineering (taking requirements as given) to agile methods (fast prototyping and including users in the process until they get it right) to very soft (drawing pictures and mind maps). All of them incorporate some level of systemness in their approaches and thus deserve to be called systems analysis. However, with what we have learned from project successes and failures using these methods we can now synthesize a much more holistic approach to the analysis of systems of all types. We are tempted to call it 'true systems analysis' in an attempt to emphasize that it includes the analysis of all aspects of a system, not just the component pieces or the information infrastructure but the whole or true system. However, we will resist that temptation, if for no other reason than to avoid having to type too many words. Henceforth, when we say systems analysis (SA) we mean whole systems will be analyzed and not just parts in isolation, and it will be deep systems analysis, not just opaque-box analysis.

The natural or physical sciences (physics, chemistry, biology, astronomy, etc.) have fared much better in terms of developing deep understanding, at least in terms of phenomena of interest (as opposed to systems of interest). The reliance on reductionist approaches has paid off in terms of exposing underlying mechanisms that account for higher level phenomena. It turns out that such methodological approaches are actually quite close to systems analysis. The difference is that reductionist methods are directed at deconstruction specifically. Systems analysis, on the other hand, explicitly seeks to maintain knowledge about structural and functional connections between the parts exposed in deconstruction. It seeks to maintain a holistic perspective of the system even as it deconstructs the parts.

4.2.2.2 A Three Phase Process

The approach to systems analysis to be demonstrated in this book is actually a three-phase process as mentioned in the Introduction. The first phase involves system identification or determining the boundary of the system of interest (SOI). As mentioned previously this is not a trivial task. It involves observation of the real system (or careful delineation of the boundary in a to-be-designed system). The analysis is of the various inputs and outputs (shown in Figure 4.1), including observation of any disturbances not directly associated with inputs from noted sources.

Once the boundary and boundary conditions have been identified, the analysis turns to the various sources and sinks in the environment that interact directly with the SOI and over its entire life cycle to the extent possible – by definition it is impossible to identify non-stationary changes that could come into play in some future time. For the noted sources and sinks, the analysis looks to characterize as fully as possible the dynamics of the flows/interactions the SOI has with the sources and sinks. As mentioned previously, the analysis does not attempt to go into the internal workings of these sources and sinks as part of the analysis of the SOI. Later we will introduce the exception to this rule when it may be discovered that the dynamics of the environmental element is not adequately described by mere modeling and when the coupling strength between the SOI and that element is such that it becomes advisable to reconsider the boundary, treating the element as part of the SOI and analyzing its internals as any other subsystem in the SOI.

That leads to the third phase which is the recursive deconstruction of the system into its subsystems and those into their sub-subsystems and so-on until a stopping condition is met. The overview of this was covered in Chapter 3; see Figure 3.5 showing 'atomic' work processes that constitute the stopping conditions for recursive deconstruction. In Chapter 5 we will examine several specific methods for deconstructing real systems based on their kind and composition (e.g. how to deconstruct a business enterprise or a living brain).

4.2.2.3 Reversing the Direction of Analysis – Enlarging the Scope

The language described in Chapter 3, based on the formal structure given in Equation 3.1, allows for an interesting approach to analysis. The basic intent is to analyze by deconstruction the SOI, which is identified as level 0, the root of the structural tree. In this way analysis is similar to the reductionist program use in the other sciences. However, there is no reason that the analysis need only go 'downward.' Since the environment has already been analyzed and logged into the knowledge base, the elements, the sources and sinks, are available. The environment, recall, is actually a supra-system on a larger than SOI scale by virtue of being connected to the SOI. That is, the original SOI is only a single subsystem of the supra-system. It is possible to reverse the direction of deconstruction by considering the boundary of a new SOI, the supra-system, pulling all of the environmental elements into the new supra-SOI.

In essence, the analyst is taking a step back to grasp the larger picture. As an example, suppose a major aircraft manufacturer wants to design a new more fuel-efficient airplane for intermediate haul flights. They have within their company all of the plane design engineers needed to accomplish this task. They believe there is a market for such a plane, that the demand will be high, and their sales will more than justify the investment in design and tooling that will have to be done. The company employs a systems engineer to bring all the pieces together for the project. But this particular engineer is 'seasoned' by experience and realizes that while there might be a market, or the appearance of one, the new airplane will introduce some major changes in the operations of the airlines and airports with respect to flight scheduling, turn-around times, and a host of other operational considerations. These would all be considered inputs and outputs to the airplane once in service, part of the environment. So, the engineer decides to widen the scope of the systems analysis to think of the SOI not just as the airplane, but the company and the customers, the airports and their operations. He calls in experts on these elements who analyze the impact of the new design on them. What they determine is that it will be quite disruptive for a number of stakeholders unless the airplane manufacturer supplies additional support equipment designs that will facilitate the use of the new airplane with minimal disruptions to other airport services.

As it happens, new airplane designs do go through a period of trial-and-error adjustment phases, the costs mostly borne by the airlines and airports. This is an example of an evolvable system. Could these disruptive adjustments be minimized by reversing the direction of systems analysis as described above? In the sciences this is done to some degree by the fact that there are specialists who work on problems at a higher level than where the reductionists are working. They are *integrators* who can use the knowledge gained by reductionists to map out the higher-level functions. They put the pieces discovered by the reductionists, of the larger puzzle together. For example, once biochemists understood what the adenosine triphosphate molecule was and saw how ubiquitous it was in the cytoplasm, people working at the whole cell level realized its role in distributing energy to the various work site organelles, like ribosomes, and how these molecules interacted with those sites.

4.2.2.4 The Science of Systems Analysis

Analysis is a process of deconstructing a system in order to find the subsystems that underlie the operations, behaviors, of the system. At its roots this is a reductionist approach, but with an important difference. Systems analysis makes a concerted effort to observe and maintain knowledge of the causal connections between subsystems. In the sciences this corresponds with the attempt to understand the context of phenomena. For example, in the analysis of the structure of the DNA molecule, known at the time to be the major molecular constituent of genes, James Watson and Francis Crick almost immediately recognized the significance of the double helix construction of the molecule and the implications of the four nucleotides (adenine, guanine, thymidine, and cytosine) forming a genetic code[6]. The deconstruction of a molecular structure was interpretable as the key to genetic knowledge because of the higher-order knowledge of what genetic information meant. Scientific deconstruction is systems analysis but not necessarily embedded in a process for recognizing the significance of a found structure or function in the context of a larger whole.

The deconstructive work of systems analysis is not much different from the reductionist program in sciences in terms of process. Both look for the sub-components and their interactions. The reductionist paradigm seeks sub-phenomena, while the process of systems analysis is always alert to the context of those phenomena as subsystems of the larger whole. The sciences have been so spectacularly successful because they have been doing systems analysis without the formality supplied by a systems approach. Until recently.

Systems analysis is the process of deconstruction of higher level processes (phenomena) to find out what sub-processes underlie the process. The main difference is the maintenance of understanding the context of the sub-processes – the system in which they are embedded. For example, cellular biology has done an incredible job of deconstructing the internal workings of cellular organelle and metabolism. The cell is the overarching system and the organelle and metabolic processes are subsystems within. This has always been 'understood' or implicit in the process of coming to understand the cell. The only difference between a systems approach and the historical approaches of cellular biology is that the latter was pursued in what we respectfully

---

[6] See the Wikipedia article: https://en.wikipedia.org/wiki/Molecular_Structure_of_Nucleic_Acids:_A_Structure_for_Deoxyribose_Nucleic_Acid for more background. Accessed 12/28/2016.

call an ad hoc framework. The mechanics and purpose of each new discovery of an organelle, like the mitochondria, was pursued primarily to see what the thing does. Only later, after working out what it did, was there a realization that what that was contributed something to the whole cell system. Eventually the systems approach was implemented as those who were able to step back and look at the integration of the bits and pieces started to put together a ‘bigger’ picture.

The current program in systems analysis is to recognize the advancements in knowledge that came from the after-the-fact integration of reductionist-derived knowledge and to turn the perspective on its head. Systems analysis is a top-down (mostly) process in which the reductionist process is the servant of the holistic or understood-to-be-integrated framework at the outset. Rather than an ad hoc deconstruction of a system for the sake of examining a bit for its own sake, the systems approach seeks to explicate the bits for the sake of understanding the whole system of interest. We believe that this change of perspective will make the sciences even more productive in their pursuits of the bits because it provides a holistic knowledge framework into which to plug the knowledge of bits as they are discovered.

In essence, all of the methodologies of reductionist and empirical science are applicable to systems analysis. The only difference is that we are starting with the intent of keeping the whole in mind as we pursue the reductionist methods. This is why systems science is a science in its own right.

Deconstruction, from a systems approach, is an algorithm for discovery. As already pointed out, a system can be represented as a hierarchical structure – a tree of systems and subsystems. The algorithm is a recursive method for discovering the child nodes in any node in the branches of the tree. That means that we reapply the discovery process at each subsystem node, finding the sub-subsystems which then become systems in their own rights. The process can be conducted in a breadth-first approach – finding all of the subsystems at any node – or in a depth-first manner – exploring a single branch of the tree until we reach a ‘leaf’ node or a sub-…-subsystem that is an ‘atomic’ unit. By ‘atomic’ we mean a system that performs some minimal process or a component whose process is already well understood. For example in decomposing a computing system we could stop when we reach the elemental logic gates since we know quite well what they do and how they do it. There are similar ‘stopping rules’ for other kinds of systems that will be covered in Chapter 5. Every recursive algorithm requires such stopping conditions in order to be useful!

The process of deconstruction (reductionist method) in the systems approach must be captured in a structured knowledgebase. The structure of this knowledgebase is determined by the principles of systems science and the formal definition given in Chapter 3. The structure of the knowledgebase is the key to exploiting the systems approach.

4.2.3 Component 2 – The Knowledgebase

A database is a set of structured files that contain data relevant to an organized system (the systems program that runs the database engine is called a DBMS). For example, an employee database contains data that is relevant to the employees of an organization such as their personal data, pay rates, employee numbers, etc. Databases are the memories of organizations, used by various programs (not the least of which is payroll) to keep track of the employees, inventories, and customers (to name a few). A knowledgebase is a database that has a more defined purpose than just keeping track of data. The main difference is that a knowledgebase is a highly integrated set of databases that constitutes all that is known and knowable about a system.

The most dominant database model used today is called a 'relational' database system. The relations between data elements is based on model schemata. For example, in an employee data base, a single file (called a table) can contain employee personal information such as address, years employed, social security number, etc. Each employee is assigned a unique employee number which acts as the 'key' to any specific employee's personal data. Tables are composed of rows for each employee number (in the key column), with the rest of the data occupying columns with headings giving the type of data contained. Another table can contain payroll data for each employee. Data such as pay rate, tax rates, withholdings, etc. are kept in rows as with the personal information table. The employee number is again used to identify each employee; it is the one thing that is common between the two tables.

The reasons for keeping these data groups separate are several, including security and ease of maintenance. In using relational DBMS for different applications, the tables are 'related' by, in this example, the employee number. Say, for example, an application is used to print payroll checks and route them to envelopes for mailing to the employee. The application first computes the pay for the employee from the data in the payroll database. It then cross references the employee number in the personal information database to set up the mailing of the check, retrieving the address, etc. Relational databases (RDBMS) engines have been perfected to do these kinds of operations using a relational language called SQL (pronounced SeQueL).

Unfortunately, a pure RDBMS is not an ideal way to represent and store systems knowledge. Systems are sets of *objects*, as established in Chapter 3, each being quite complex. Objects are whole things that are not represented easily in RDBMS tables. Fortunately, there has been a growing interest in object-oriented databases (OODBMS) in which the storage element is an object, and the object can have considerable complexity. At present the field of OODBMS is not as developed as RDBMS and so the ability to readily store system object knowledge is still in a state of flux. However, the major outlines are clear. In Chapter 7, in Part 2, we will see examples of the way in which we will capture the knowledge from systems analysis and store it in a way such that it is searchable and 'relatable' to other objects (e.g. subsystems with interactions).

4.2.4 Component 3 – Modeling

The most active area of using systems theory in the sciences and engineering is the building of system models for simulation in a computer (as indicated above). Indeed, many of the sciences have adopted subfields called ‘systems <science name>’ based primarily on the use of simulation modeling. Unlike earlier mathematical models involving differential equations, where higher-order or non-linear differential equations proved intractable with respect to closed form solutions, simulations can involve arbitrary durations and numbers of internal feedbacks. Numerical methods are used to approximate within nearly arbitrary precision what a formula based on continuous (real number) methods in the calculus would produce if solvable.

All models are abstractions of the real systems they attempt to emulate. The construction of a model involves artful choices by the modeler regarding the ‘boundary’ (conceptual in this case) and the granularity of internal components. They must also choose what they believe to be relevant variables to monitor as the simulation run progresses. This is called ‘instrumenting the model’ and refers to the periodic recording of the variables of interest in order to collect data to be used in dynamical analysis. The reliance on the pre-existing knowledge of the modeler, plus their experience in constructing meaningful models (based usually on having constructed not-so-meaningful models earlier in their careers!) makes the part of systems science more of an art than a science. The approach works to some degree in engineered systems development, in part because there is already a rich library of system models (templates) that have been proven in the past and can be used to guide new designs. Thus, it is more knowledge-based than happens in the sciences when exploring phenomena.

We have argued that, traditionally, the use of models has been based on having a lack of transparent-box knowledge and or an inability to observe and measure internal mechanisms of a system. The models substituted for that kind of knowledge by using informed best guesses as to how a system worked to construct the model. If the simulation (solution) of the model then produced results effectively similar to the behavior of the real concrete system, the analysts felt justified in concluding that their understanding was sufficient to use the model for its real purpose – prediction. In the process being set forth here we assert that more analysis (deconstruction or transparent-box analysis) be done first before constructing models. That means a model is less built as a crutch to understanding as a tool for confirmation of knowledge already gleaned from the analysis. At the very least this reversal of roles means that models used for prediction purposes should have higher levels of confidence associated with their results.

In Chapter 9, on modeling, we will show how a model derived from a more complete transparent-box analysis increases our confidence that its predictions are more veridical and can be brought into play for that use more quickly than is currently the case. In the Box 4.1, “Getting the Horse in Front of the Cart,” we explain how modeling has been used as a substitute for deconstruction analysis in the past, and how our sciences are changing to warrant not using this approach.

Model generation is based on the fact that all of the relevant details of structure and function of a system at any arbitrary level of organization has been captured in the knowledgebase and is available for use. The model itself is a replication in, say, software code very similar to a system dynamics model in Stella or similar environment. A group of software tools can be developed that draw from the knowledgebase and generate the code. Alternatively, and in support of the code for simulations (see below) tools such as SysML might be developed to generate conceptual models (diagrams) for visualization of the system at a particular level of organization (i.e. abstraction). These kinds of tools and their uses will be covered in the relevant chapters to follow.

Box 4.1. Getting the Horse in Front of the Cart

Throughout the history of the natural sciences we have been faced with the problem of not being able to see into the interiors of objects of studies. The boundaries that demark an object from its environment tend to be opaque and/or the subsystems that comprise the internals of the object are too small to observe directly. Over time we have invented a variety of instruments that have allowed us to penetrate boundaries and observe ever smaller subsystems, e.g. high-power microscopes to see cells or particle beams to see inside atoms. At first, the use of these instruments largely destroyed the integrity of the entity being observed if they had to penetrate the boundaries. Thus, there have been extreme limits on the ability to convert an “opaque box” into a transparent one in which we can see the internal relations and functions in real time.

The method developed by the sciences could be described as partial deconstruction (usually with destruction), where deconstruction was even feasible, followed by guesswork as to how the parts really interacted in an intact system. That included constructing models of the systems’ internals from those guesses and solving the equations, or running the simulations to see if the modeled system behaved the same as the animate real system. Our knowledge of the system depended on how well we guessed and how closely our models replicated the essential features of the real system. Thus, modeling is viewed as a tool or method for coming to understand a system when it is assumed to be non-deconstructable without damage. This approach fits neatly within the general framework of the scientific method – observe a phenomenon as best you can, guess as to how it works (generate hypotheses), build a model and see if its large-scale behavior matches that of the real system, if it does you have failed to disconfirm the hypotheses. Or, in other words, you gain increased confidence that your model represents the real system. Run the model under as many conditions as you can observe in the real system and continue to confirm the match. But, if the model system fails to behave under some conditions, you go back to the drawing board and make some additional guesses, tweak the model, and try again.

What generally happens in the natural sciences, however, is you return to the observation stage and attempt to gain more information about the internals. This is a workable strategy as long as you have many disposable samples of the system to dissect. If you only have one system, like the economy, then the approach to further deconstruction can be tricky. For example, a key component of any economic model is human decision making. Since it had not been possible or even feasible until quite recently to deconstruct the mind (brain) the use of ‘agent-based’ models in which the decision processes are hypothesized has been the standard approach.

We have entered an era of a whole new approach to instruments for deconstructing systems. Advances in technology now allow us to see inside many different kinds of systems without disrupting their functions, and at many different scales of resolution. What this means for systems analysis is that our need to rely on modeling to test ideas about what must be happening inside a system is lessening. In other words, the process of converting an opaque box into a transparent one can be approached more directly as these instrumentation technologies improve. One of the best examples of this is the development of functional magnetic resonance imaging (fMRI) used to directly observe biological processes as they happen in real-time. And a dramatic use of fMRI has been observing what is going on inside the brains of human beings as they make various kinds of decisions. We will explore this development and what it means for systems analysis in Chapter 5, illustrating how it has become possible to do deep systems analysis in a variety of social system domains involving human decision making and emotional processing. The bottom line is that psychology, the science of the mind, and neuroscience, the science of the brain, are forging an operational alliance that gives us powerful tools for doing deep analysis without the need for generating hypothetical models first. In essence, models will be seen to be more reliable tools of prediction, more so than tools of exploration. We are getting the horse back in front of the cart where it belongs.

4.2.5 Component 4 – Simulations and Hypotheses Testing

Once a model of a system, at whatever level of organization, is available it is available for running in simulation. The code generated in Component 3 above can be run on a computer, after determining appropriate time steps. The same visual framework used for analysis and data capture can now be used to animate the simulation. Output from the simulation, the data recorded for the variables of interest, can then be analyzed and graphed appropriately just as is the case now. Many of the tools needed for this component already exist.

4.2.6 Component 5 – Generating Designs or Policies

In the case of engineered systems, the ability to generate a set of design and performance specifications is essential to success. In the current approach to doing this the process is labor intensive as various specialist engineers have to analyze the requirements and apply domain-specific knowledge (e.g. electrical engineering) to generating designs for some specific functionality within a subsystem. Engineers are expensive people to employ. Moreover, the production of truly qualified engineers in many fields is lagging sadly behind the needs of modern society. Learning to be an engineer is hard work and many modern students are not inclined to pursue this line of education[7].

Today we have a significant amount of knowledge of how to produce design specifications given that we have well defined functional specifications. It is entirely conceivable that once a to-be-designed system is captured in the knowledgebase (i.e. we have an in-depth description of

[7] Sadly, even those who do wish to major in an engineering degree are often not adequately prepared for the rigor of the field or they have been given a very superficial orientation to what is involved in it. Our modern education system puts emphasis on the fact that engineering professionals make good salaries and uses that as a motivation to get students interested in the fields. Then once they start taking the courses needed to obtain the degrees they realize how hard it is. Motivating students to take difficult subjects based on future potential salary rewards is probably not the best way to increase the number of engineering graduates.

the structures and functions of systems and subsystems, etc.) the process of generating design specifications can be readily automated and easily checked[8].

This is actually a corollary result to the generation of models discussed above. A design is, after all, an abstract version of a concrete system, containing meta-data regarding the construction of components from the lowest level of organization up to the highest level. In other words, the same knowledge that is captured in a knowledgebase that allows us to generate models is the exact knowledge needed to generate design specifications. In fact, in engineering the typical approach is to co-generate specifications along with models used to test the designs being proposed. The two processes are Siamese twins co-joined at the hip, as it were.

4.2.7 Component 6 – Monitoring Behaviors

This subsystem reflects the reality of complex adaptive and evolvable systems. Every such system must monitor its own behavior (state changes over time) relative to the desired results. In this case a monitoring subsystem is in place to determine whether or not the process is producing reasonable results. The main method for doing so involves measuring the outputs of several other subsystems (components) such as the modeling, simulation, and design specifications/policy generation processes. These are all the ‘action’ outputs from the entire process (like the movement outputs of brain decisions processing). It has to be determined that they are producing results consistent with the objectives of the systems understanding process as a whole or, otherwise, there have to be governance interventions in the process to direct it toward ‘better’ outcomes. The methods of monitoring (measuring and comparing to goals – standard first-order cybernetics) will be developed further in Part 3.

4.2.8 Component 7 – Process Governance

Chapter 11 is devoted to the architype model of sustainable CAES called ‘governance’. This pattern is ubiquitous in all CAESs in nature and is so important to understanding why systems persist over time that it deserves a whole chapter. As with all such systems, the system of system understanding has its own governance sub-process to ensure it gets the job done right! Those involved in various kinds of systems processes today will recognize parts of this sub-process, for example as “project management.” But viewed from a holistic systems perspective it is so much more.

One immediate implication of a governance process is not only to obtain and maintain stability in the system process, but also to manage the expenditures of resources (costs) as compared with the receipt of resources (revenue or benefits). We outline that function next.

[8] For example there is a growing body of knowledge in what is called model-based design in which tried-and-true models of designs for ‘paradigm’ systems serve as templates for developing designs of specific systems matching the template.

4.3 Cost-Benefit of the Process[9]

The systems understanding process is undertaken within the context of how much benefit does society or a profit-oriented organization gain from it relative to the costs incurred in doing it. The process obviously involves both direct costs and risks of additional costs if something goes wrong. So, the main question everyone will ask is: Does the benefit of deep understanding gained via this process exceed the costs involved in undertaking it? It's a fair, but greatly misunderstood question.

The current prevailing concept of economics has led managers and investors to focus attention on the short-run. Quarterly performance seems to be the main decision factor in working out budgets for investments. The sentiment seems to be: "Take care of quarterly earnings and the long-run will take care of itself." There may be little that can be done at present to change this sentiment. The economic theories of neoliberal capitalism provide internal positive feedback loops that reinforce it.

However, here we present some arguments against this sentiment or, more correctly, for a different sentiment that puts more value on long-run thinking. We will return to this theme in Chapter 8, "Analysis of the Biophysical Economy," after demonstrating that a systems approach to studying the economy exposes some of the fallacies lying behind the neoclassical theories and the neoliberal capitalism that dominates the global economy today. In anticipation of many arguments, we acknowledge that this may be a quixotic quest for now. But, at some point in the not-too-distant future we predict there will be a major disruption of the global economy as a result of the short-term, profit-oriented focus. It is that focus that leads to taking shortcuts in systems projects (e.g. mega-projects). As already pointed out, the failures of these projects are already costing billions of dollars lost to failures of various kinds. This portends future problems as projects only get larger and more complex.

In this section we review various costs and benefits associated with using the deep understanding process just outlined and compare them with those associated with shallow understanding.

To begin with, a deep understanding process will be as costly as the complexity of the system we seek to understand. That is, the costs, roughly indexed in human labor hours, increase with the complexity of the system, as we have defined that metric in Chapter 2. Moreover, though there is no solid empirical evidence to support this, the intuition is that the relation is nonlinear, possibly exponential and almost certainly quadratic at least. The reason is the overhead needed to consider all of the subsystems and their relations for systems that have

[9] In this section we will be referring mostly to business and government projects, however, it should be noted that most of these factors apply equally to large science projects such as The Human Brain Project, which seeks to build an exascale supercomputing platform for research support of models of the human brain at multiple scales of resolution. See the Wikipedia article: https://en.wikipedia.org/wiki/Human_Brain_Project for background. Accessed 1/15/2018.

greater hierarchical depth seems to expand with that depth. The number of interactions and especially the communications channels (message flows) needed to coordinate more subsystems appears to increase in this fashion. This is supported by a simple result from graph theory when a graph is generally dense, meaning have connections between most of the nodes with each other. In a simple complete bidirectional graph the number of edges, N, increase as $n * (n - 1)/2$, n being the number of vertices, which is order n^2. However a system's internal subsystem connections are not simple unidirectional graphs. Most interactions are flows or directional, and accompanied by bidirectional communications which increases the number of edges at least by two. Thus, the more subsystems found in any level of the system, the longer it will take to analyze the connectivity.

The deep understanding process requires a thorough analysis of the subsystems and their interconnectivity at every level. The approach most often taken today is to make some assumptions about systems based on prior experience with similar systems. That is, analysts avoid doing a deep analysis by assuming that a system is comprised of subsystems and interconnections as seen in other (presumably) similar systems. This saves a lot of time. If the assumptions are correct then one need merely copy the architecture/design of the model system and consider that sufficient for making claims about the nature of the current SOI. The problem is that all too often they are not the same. Even small differences can make disproportionate large differences in the target system. Nonlinearities in subsystem functions (such as a misplaced amplifier) can lead to unintended behaviors, among many other things that can go wrong in blindly copying what is assumed.

For one thing the model system being used to form these assumptions may not be correctly modeled in the first place; it having been, itself, the result of assumed designs (or hypotheses in the case of scientific decomposition). It simply is the case that when it comes to actually understanding a system there is no substitute for the kind of deep analysis covered in Chapter 5. Only after having performed that kind of systems decomposition can one be confident that the models and sub-models are veridical.

But, time is money. The longer it takes to get the analysis right, the longer it takes before a reward is realized. Given the belief in the time value of money, in our modern capitalist enterprises[10], this amounts to diminished returns on investments so there is little incentive to 'waste' much time. Coupling this with the over-confidence of most people, scientists and engineers, in their abilities and expertise, you have a toxic formula for disaster.

So, there is a compelling reason for not doing a thorough analysis of the SOI as outlined in Chapter 5. It costs too much. Similarly there are high costs associated with collecting the data and maintaining an adequate knowledgebase as described in Chapter 7. So-called knowledge

[10] It is understandable that this attitude arises in engineering projects within for-profit organizations or non-profit ones where limited resources means controlling costs. Sadly, however, the same kind of thinking has entered the realm of the sciences where grant monies are bid upon and 'useful' products are often the goal of the research.

management can be very expensive in time and labor. So, there will be a tendency to do the minimum in the realm, for example, of model building (see Chapter 14) or generating detailed specifications for designs.

The bottom line (pun intended) is that doing deep systems understanding has a high up-front cost on the analysis phase. There is also the possibility that total costs will be very much higher. Each of the phases as described in this chapter probably has additional costs associated with the level of effort required. And this alone would dissuade organizations from adopting the process in order to gain deep understanding.

But.

What about the long-term costs?

What does it cost over the life of a complex system to operate it if the design is buggy or inefficient? What does it cost if the system fails to deliver its promises?

The kinds of complex systems that are the targets for today's considerations, e.g. complex social systems, or the Internet of Things, or ecosystems to be managed, entail significant costs if something goes wrong because some little part of the system was not sufficiently understood. In other words, the cost of not getting it right potentially far exceed the costs of doing the project well or not getting to market rapidly.

Take the argument for the latter concern. If we don't get this project done quickly someone else will. We'll miss our opportunity. We'll miss getting income right away.

So what?

What is the benefit to anyone of putting a product out there that is doomed to failure due to un-understood design? Who profits? The case studies abound where organizations of all stripes pushed a project through only to find their long-term profitability (or cost minimization) over the life cycle of the project/product usage suffered tremendously.

What, then, of benefits? They have to exceed costs in order to make the project worthwhile.

That is actually part of the systems analysis. It is undertaken in the first place because the existence of a system to perform a function that produces X in benefits, is perceived as solving a problem that currently costs Y and $X >> Y$ is the goal. What should be invested in the project depends on how much this relation produces and over what time scale. In the case of sciences the measures of X and Y are not as clearly defined – what is the value of new knowledge? In the case of an engineered solution it should be possible to estimate the relative values. In engineering it is possible to continue to test the proposition that $X >> Y$ as each stage of the process proceeds. If at some point it appears the proposition is not valid or, at least, in question, then it is possible to stop further developments and cut losses of to-date investments. It should be noted, however, that the ability to abandon sunk costs often takes a level of wisdom that is not particularly granted to many managers!

4.4 Conclusion

The process outlined in this chapter is a system for gaining deep understanding of any system of any complexity. It consists of seven component sub-processes all of which are required to do good job of understanding the system of interest. Details of these sub-processes will be presented in the rest of this book. Since the analytic process is easily the most important component, Chapter 5 will go into great detail as to how this process works. Chapter 6 will provide an example of following the analysis process and capturing the data relevant to the definition of system from Chapter 3. Then Chapter 7 will elaborate on the structure and functions of a knowledgebase which is built from the data collected in analysis and organized according to Equation 3.1 and subsidiary equations in Chapter 3.

The process of gaining deep understanding of complex systems is, unfortunately, expensive as compared with how systems are analyzed and specified today. The reason has to do with the drive of profit-oriented capitalism and market competition – we rush to get things done as quickly and expediently as possible. But this leads to errors in analysis and judgements of efficacy that cannot be caught at an early stage; they must be experienced as failures to be detected.

That mode of engineering and building complex systems entails even higher costs to society and users in the long run. In truth we don’t really have a good idea of the total costs to society of these kinds of failures. But in many case studies where short-term vs. long-term costs have been tracked, invariably the long-term costs of poor understanding have far exceeded the short-term costs even after taking into account the time value of money.

As humanity seeks to push the envelope of complex systems or seeks to better understand complex systems such as the whole Earth ecology, it will find that trying to do so profitably, i.e. benefits outweighing the costs, will require a rethinking of the upfront effort, admitting higher costs of deep analysis, but gaining much higher returns on investment in the long term.

4.5 References and Further Reading

Checkland, P (1999) *Systems Thinking, Systems Practice, Includes a 30-year Retrospective*, John Wiley & Sons, Ltd., Chichester UK.

Mobus, G & Kalton, M (2015) *Principles of Systems Science*, Springer, New York.

Rousseau, D, Billingham, J, Wilby, J, and Blachfellner, S (2016) "In Search of General Systems Theory", *Systema* 4(1) Special Issue - General Systems Transdisciplinarity: 76-99.

Troncale, L (1978), "Linkage Propositions Between Fifty Principal Systems Concepts," in *Applied General Systems Research: Recent Developments and Trends: N.A.T.O. Conference Series II*. *Systems Science* (G. J. Klir, Ed.) Plenum Press, N.Y., pp. 29-52.

Von Bertalanffy L (1969). *General Systems Theory: Foundations, Development, Applications*, George Braziller, New York.