Evaluating Engineering System Interventions

Wester C.H. Schoonenberg, Amro M. Farid

Abstract—**Our modern life has grown to depend on many and nearly ubiquitous large complex engineering systems. Transportation, water distribution, electric power, natural gas, healthcare, manufacturing and food supply are but a few. These engineering systems are characterized by an intricate web of interactions within themselves but also between each other. Furthermore, they have a long-standing nature that means that any change requires an intervention into a legacy system rather than a new "blank-slate" system design. The interventions themselves are often costly with implications lasting many decades into the future. Consequently, when it comes to engineering system interventions, there is a real need to "get it right". This chapter discusses two types of engineering system interventions; namely those that change system behavior and those that change system structure. It then discusses the types of measurement that can be applied to evaluating such interventions. More specifically, it contrasts experimental, data-driven and model-based approaches. It recognizes that only the last of these is appropriate for interventions that change system structure. Consequently, the chapter concludes with a taxonomy of engineering system models including graphical models, quantitative structural models and quantitative behavioral models. The chapter concludes with a discussion of promising avenues for future research in the area; namely hetero-functional graph theory and hybrid dynamic systems.**

Index Terms—**Engineering Systems, Intervention, Measures, Evaluation, Performance, Effectiveness**

I. Introduction

This chapter provides an overview of the background, context, and theory for evaluating engineering system interventions. So far, the Engineering System Design Handbook has provided: (1) background and motivation for the engineering systems approach, (2) theory for describing engineering systems, and (3) an overview of intervention design for engineering systems. This chapter concludes the third part of the handbook with the background and framework to support the evaluation of engineering system interventions. The first section defines a point-of-departure for this chapter to enable the study of the chapter as an independent work. The section also refers to other chapters in the book to provide context and other relevant material.

A. The emergence of engineering systems

In the context of 21^{st} century grand challenges, the field of engineering systems has emerged at the intersection of engineering, management, and the social sciences. Over the past decades, engineering solutions have evolved from engineering artifacts that have a single function, to systems of artifacts that optimize the delivery of a specific service, and then to engineering systems that deliver services within a societal and economic context. In order to understand engineering systems, a holistic approach is required that assesses their impact beyond technical performance. Engineering Systems are defined as:

Definition 1. Engineering System [1] A class of systems characterized by a high degree of technical complexity, social intricacy, and elaborate processes, aimed at fulfilling important functions in society.

Furthermore, there are a number of characteristics that distinguish engineering systems from other systems. Engineering systems...

- ... exist in the real world. They always have physical components, but are also likely to contain informational components.
- ... are artificial. Engineering systems are man-made, but often integrate into the natural world.
- ... have dynamic properties. Engineering systems change over time, and have a sense of temporality.
- ... have a hybrid state. The states of engineering systems are usually both discrete and continuous.
- ... contain some human control.

Some types of systems with these characteristics include electric power grids, transportation systems, healthcare delivery systems, the energy-water nexus, etc. This list is far from exhaustive and the reader will find numerous other examples throughout the handbook.

The growth of engineering systems has been mostly organic and incremental. Many of these systems have been expanded or shrunk to match the changing (or perceived) needs over time. This has caused inefficiencies and unforeseen dynamics within those systems. The successful implementation of engineering system interventions relies on rigorous evaluation for a future-proof design.

B. The importance of evaluating engineering system interventions

As the complexity of engineering systems has evolved, there is need for a deeper understanding of the design and operation of engineering systems. Interventions were often designed and implemented with merely a theoretical understanding of their impact. During the

Thayer School of Engineering, Dartmouth College, Hanover, New Hampshire, USA, Wester.C.H.Schoonenberg.TH@Dartmouth.edu
Thayer School of Engineering, Dartmouth College, Hanover, New Hampshire, USA, amfarid@dartmouth.edu

rise of the automobile, many transportation infrastructure systems were overhauled to facilitate this new mode of transport [2]. However, the impact and outcomes of such changes were often unforeseen by policy makers. Today, we have the ability to much more accurately evaluate and understand the impact of interventions in engineering systems. Increased data and computational resources enable evaluation of interventions both *after* and *before* implementation [3].

The evaluation of interventions after implementation was especially useful when the computational resources to predict the outcomes of interventions were limited. The results of previous interventions guided new interventions and were eventually generalized as "rules-of-thumb". Furthermore, this type of evaluation also helped "tune" (or salvage) the intervention to get the best results. The downside of this trial-and-error approach is obvious; sometimes the interventions do not perform as intended. The failure to perform may become immediately obvious, but can also materialize when the system has to function under extreme circumstances [4]. An example of the latter scenario is the failure of the electric rail system in New York City during hurricane Sandy. The rain ahead of the hurricane flooded several transformers and the power supply for the rail system was interrupted. As a result, the rail system failed and evacuation of lower Manhattan was severely interrupted. It took a major hurricane to demonstrate the limitations of the electrification of the rail system, whereas it had been operated successfully for decades before.

In order to improve interventions before they are implemented, evaluations are now often performed *before* they are implemented. The goal is to determine if the intervention will improve the system outcomes before a large investment is made and society has been interrupted [5]. Furthermore, predictive evaluations can be used to evaluate if the improvement is large enough to outweigh the downsides of the intervention. One of the challenges is to accurately represent the real-world system with computer models. When the system has not been represented accurately, an intervention can work well in the simulations, but may under-perform in reality. In theory, the more extensive the model, the more accurate the prediction, but in reality this is often not feasible due to the financial constraints to build a complex model and the computational constraints to simulate it. Therefore, a balance between simplifying and detailed modeling is required.

When it comes to engineering systems interventions, there is a real need to get it right. Engineering Systems are inherently socio-technical, they impact and are impacted by people. Furthermore, they are expensive to build and change. This chapter discusses both the predictive as well as the post-implementation evaluation of engineering system intervention.

C. Relation of this chapter to the previous chapters

Placeholders for the editor. TBD.

- Why this handbook?
- Why this chapter?

D. Chapter Outline

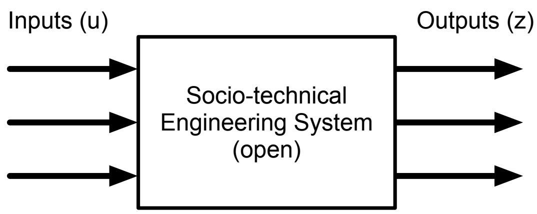
- **What is an Intervention?** In order to evaluate engineering system interventions, first the word "intervention" must be understood. Engineering systems are most often legacy systems, and any change to the system is inherently an intervention of some type. Section II discusses systems and the different types of interventions.
- **Evaluation Requirements.** Artifacts, systems, and other things can be measured in one of two ways. (1) Direct measurement and (2) indirect measurement. These two types of measurement bring along their own set of specific requirements in order to result into a holistic and appropriate evaluation of the engineering system intervention. Section III discusses the the fundamentals of measurement and their application to engineering systems.
- **Comparing Evaluation Methods.** Evaluation of engineering systems interventions requires a deep understanding of the impact of the intervention on the outputs of the system. Section IV provides an overview of three approaches to intervention evaluation methods.
- **Model-Based Intervention Evaluation.** Finally, the chapter concludes with a discussion around the importance of data and systems theory in the evaluation of engineering systems. Model-based evaluation approaches leverage theory to enable intervention evaluation for engineering systems. Section V provides an overview of some of the most important system modeling methods.

II. What is an Intervention?

This section introduces a holistic understanding of the meaning of *interventions*. This is realized by first summarizing the description of a system to define a consistent framework. Based on this framework, for the purposes of this chapter, a definition of *interventions* and a discussion around the types of interventions are provided. The type of intervention is critical when making a decision about the type of evaluation method, as discussed in Section IV.

A. Describing Systems

Engineering Systems, also referred to as socio-technical systems, are complex systems at the intersection of physics, management, and social sciences [1]. The evaluation of engineering system interventions relies on accurate and consistent measurement of the system. As shown in Figure 1, this chapter adopts the approach of many STEM disciplines where systems are mathematically described as a system of differential algebraic equations (DAEs) that define the relationship between the inputs u and the outputs z [6], [7]. The system is also said to have states x, algebraic states y, and parameters λ. The vector functions $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are differential equations, algebraic equations, and output equations respectively. While a more complex model based upon the hybrid dynamic system literature is possible, a system of differential algebraic equations serves the purposes of this discussion.



$$\dot{x} = f(x, y, u, \lambda)$$
$$0 = g(x, y, u, \lambda)$$
$$z = h(x, y, u, \lambda)$$

Fig. 1. A mathematical and graphical representation of an arbitrary engineering system.

In addition to the above description, this chapter requires the introduction of four systems thinking abstractions: (1) system context, (2) system function, (3) system form, and (4) system concept [1], [8]. These abstractions support the classification of intervention types and their accompanying evaluation methods.

1) System Context: The system context is the set of interrelated conditions in which the system exists or occurs [9]. Sometimes, it is also referred to as the system environment: *"All that is external to the system"* [10]. The field of Engineering Systems emphasizes that the system does not operate in a vacuum, but rather is solidly placed in its context. When an intervention is evaluated, the impact of the system on the context is critical to truly understand the system's performance. Sometimes, these outputs are neglected with severe consequences (e.g. climate change). Naturally, the context also influences the system itself and often, it determines the success of the intervention.

2) System Behavior: System behavior is the response of system outputs to a change in system inputs or parameters. It reflects the processes, or function of the system: "what the system does". The system inputs are predominantly a result of the system context, whereas the parameters are internal to the system. In the context of *Engineering Systems*, the system behavior consists of the behavior of the engineering artifacts and the humans that interact with the system.

3) System Form: System form is the description of a system's component elements and their relationships. The system structure also defines the presence (but not values) of system states x, algebraic states y, parameters λ, inputs u and outputs z. By adding or removing elements to/from the system, the number of equations in the vector functions $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ changes.

4) System Concept: The description of the system as a whole relies on the combination of the system behavior, and the system structure. System concept is the mapping of system function onto system form (also called the allocated architecture [11]). Consequently, a system of equations can represent system concept. The behavior of the system results from the coupled equations.

B. Describing Interventions

For the purposes of this chapter, *"intervention"* is defined as:

Definition 2. Intervention: [9] The act of interfering with the outcome or course especially of a condition or process (as to prevent harm or improve functioning).

In the context of engineering systems, interventions intend to change the system so as to improve the outcome of the engineering system. Two types of interventions are recognized: behavioral and structural.

1) Behavioral Interventions: Behavioral interventions aim to change the outcomes of a system by adjusting the values of the system inputs and system parameters while the structure of the system is untouched. As a result, behavioral interventions are often relatively affordable. Decisions to change the operating procedure or policies around a system may take a long time and are sometimes hard to implement, but the upfront capital investment is limited because no fundamental changes in the system are necessary.

An example of an intervention based on system inputs is a policy change that increases the ethanol percentage in gasoline. When a different ethanol/gasoline mixture enters the system, the emissions of the transportation system will change as a consequence.

An example of an intervention based on system parameters is the reduction of ticket prices in a public transit system. Ticket prices are internal to the engineering system and are set as a result of a policy decision. As a consequence of this parameter change, the total public transit ridership may increase/decrease, with cascading impacts such as: less/more traffic, less/more emissions, etc.

2) Structural Interventions: Structural interventions aim to change the structure of the system; its parts and the relationships between them. These changes are often physical and require large upfront capital investments. Furthermore, structural interventions require a revision of the operating procedures and policy around the system, since the policies of the old system may no longer apply.

An example of adding elements to the system is the addition of a road in a town. This road adds an "equation" and a "state." For example, a description of the traffic flow on the road as a result of the number of vehicles on the road. Such a structural intervention leads to a revision of the local traffic ordinances. For example, at the connecting intersections a new speed limit may be introduced to reduce the risk for turning vehicles.

An example of adding variables is the consideration of electric vehicles for parking lot design. Electric vehicles require charging facilities on the parking lot, which changes the calculation of the required parking spots in building code.

This section first introduced a common framework for describing systems using four systems thinking abstractions. These distinct abstractions are the basis for the selection of the appropriate method for evaluation of the intervention (to be discussed in Section IV).

III. Requirements for Evaluating Interventions

This section discusses measurement as a foundation for the evaluation of engineering system interventions. Interventions aim to improve the existing engineering system. Consequently, the evaluation of interventions requires a comparison of (at least) the current system and the system with the intervention. Such a comparison requires the definition of a common mathematical framework (or standardizing space) to describe both systems. The process of first defining this framework and then describing the systems within the framework is called *"measuring."*

This section first discusses the fundamentals of measurement including an overview of the generic measurement process, measurement scales, and different measurement strategies. The second part of this section then discusses different approaches to measurement, and specifically, the differences between measuring a technical system and an engineering system. Based on this foundation in the measurement of engineering systems, Section IV discusses the evaluation methods for engineering system interventions.

A. Measurement Fundamentals

The measurement of engineering systems is critical for informed decision-making. As shown in Figure 2, without measurement, the real world presents us with an empirical system that exhibits certain phenomena called empirical results. These results can be viewed as qualitative or anecdotal evidence. Nevertheless, the link between the empirical system and its empirical results is often not well-understood and consequently the associated intelligence barrier prevents effective decision-making. Instead, the empirical system is first *measured* so that real-world phenomena are assigned their associated numerical values in a formal (mathematical) system. Mathematics, and statistics more specifically, are then used to determine numerical results in the formal system. These are, in turn interpreted, to become empirical results. Without an accurate and consistent approach to measuring the empirical system, the foundation for the decision-making process is flawed. Consequently, the empirical and formal systems must posses methods by which their respective objects can be *related* and ultimately compared.

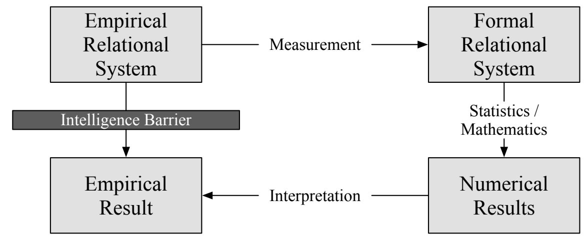


Fig. 2. A Generic Measurement Process [12]

More specifically, the **Empirical Relational System** contains a nonempty set of empirical objects that are to be measured, with relations between and closed binary operations on the empirical objects. Note that these relations are *independent* of the measure function. The **Formal Relational System** is a nonempty set of formal objects with relations between and closed binary operations on the formal objects.

Definition 3. Measurement [13], [14]: "Measurement is the process of empirical, objective assignment of symbols to attributes of objects and events of the real world, in such a way as to represent them, or to describe them." – Finkelstein, 1982

Measurement consists of three elements: (1) a set of measurables, (2) a standardizing space, and (3) a measure function. The set of measurables is defined as a set of objects with a specific attribute type. The standardizing space is a basic construct to which all the mesaurements can be compared. Finally, the measure function performs the empirical and objective assignment as mentioned in the definition of measurement. A consistent measure function ensures a consistent measurement of empirical relational systems to formal relational systems. If two empirical systems have been translated to formal systems with the same measure function, the formal systems can be compared rather than their respective empirical systems.

Definition 4. Measure [12]: A measure (or measure function) is a one-to-one function that acts on a set of (empirical) objects and returns a formal object.

Note that often the term "measure" and "metric" are confused. Metric, however, is defined as:

Definition 5. Metric [15]: A metric, also called a distance function, defines the distance between a pair of elements in a set.

Not all empirical relational system can be measured in the same way. For example, human behavior and a block of iron do not have the same attributes. The type of empirical system, with the related attributes, determines the type of measurement scales that can be used to measure the system. This impacts the type of numerical results downstream in the measurement

TABLE I
Classification of measurement scales [12]

Scale Type	Applicable Statistics	Example
Nominal	Non-parametric	Football player uniform numbers
Ordinal	Rank Order & above	IQ
Interval	Arithmetic Mean & above	Celsius Scale
Ratio	Percentage & above	Kelvin Scale
Absolute	Additivity & above	Counting

process, because not all mathematics and statistics can be used for all measurement scales. The scale types, with applicable statistics and examples are presented in Table I. Engineering systems inherently combine physics-based systems with human behavior and economics. Consequently, the measurement of the engineering system requires a combination of the measurement scales.

From a practical perspective, there are two measurement strategies: (1) direct measurement and (2) indirect measurement. Direct measurement is applied when the desired property is both "simple" and an "output" of the system. As a result, the property is easily accessible and there are often sensors that directly convert the desired property into a numerical result. Fundamental measures like length, time, voltage, and current are examples. However, these properties are rare, especially for engineering systems. Indirect measurement applies to properties that are not fundamental. These properties require the combination of fundamental properties, that are considered "internal" to the system, into a formal model. The formal model is considered the standardizing space and mathematics and statistics are applied to this model to extract the desired numerical results.

B. Engineering System Measurement

During the past century, engineering solutions have evolved from engineering artifacts to engineering systems. Consequently, the solution requirements have changed. Instead of merely "functioning" artifacts that performed their (singular) task, engineering systems perform many services composed of separate tasks. Furthermore, engineering systems include non-technical elements, more specifically humans. It is, therefore, essential to evaluate engineering systems beyond their technical aspects and include impacts of the system on its environment.

This section describes Engineering System measurement with a tiered approach. Engineering Systems are evaluated at several levels of granularity. First, the fundamental artifacts are evaluated based on the performance of their specific task with Technical Performance Measures (TPMs). Then, the combination of these artifacts provides a service. The performance of these services is measured with Measures of Performance (MOPs). The first two types of measures, however, do not truly address the socio-technical nature of Engineering Systems. Therefore finally, Measures of Effectiveness (MOEs) were developed at the highest level of granularity for Engineering Systems. These consist of multiple services and socio-technical interfaces. For the Engineering Systems literature, a subset of these measures is especially important; *life cycle properties* or *ilities*.

Definition 6. Technical Performance Measures [16]: "TPMs measure attributes of a system element to determine how well a system or system element is satisfying or expected to satisfy a technical requirement or goal."

Definition 7. Measure of Performance [16]: "The measures that characterize physical or functional attributes relating to the system operation, measured or estimated under specified testing and/or operational environment conditions."

Definition 8. Measure of Effectiveness [16]: "The operational measures of success that are closely related to the achievement of the mission or operational objective being evaluated, in the intended operational environment under a specified set of conditions; i.e., how well the solution achieves the intended purpose."

Overall operational success criteria (Measures of Effectiveness) include: Mission performance, safety, operability, operational availability, etc. These measures of effectiveness are often a quantitative means of measure a degree of adherence to requirements.

Finally, in the context of engineering systems, life cycle properties or *ilities* need to be addressed as a subset of the MOEs. The definition of ilities is:

Definition 9. "ilities" [1] "The ilities are desired properties of systems, such as flexibility or maintainability (usually but not always ending in "ility"), that often manifest themselves after a system has been put to its initial use. These properties are not the primary functional requirements of a system's performance, but typically concern wider system impacts with respect to time and stakeholders than are embodied in those primary functional requirements. The ilities do not include factors that are always present, including size and weight (even if these are described using a word that ends in "ility")."

This section described (1) how to measure and (2) what to measure. The former was described through the process of measurement and the latter was described through three categories of engineering system measures in increasing scale. The chapter now builds on this knowledge to compare evaluation methods for Engineering Systems.

IV. Comparing Evaluation Methods

This section discusses the different types of evaluation methods for engineering system interventions. As discussed in Section II, Figure 1, engineering systems create a relationship between inputs and outputs. The interventions aim to improve the outputs of the system, given a set of inputs. The goal of the evaluation methods is to predict how an intervention changes the outcome of the engineering system. Generally, the relationship between inputs and outputs of systems have been studied using one (or a combination) of three approaches.

The experimental approach was used at the origin of science. In this approach a hypothetical relationship is tested through a set of experiments in which either the inputs are changed, or the system is changed [17]. The experimental approach is generally performed in a controlled environment.

With the rise of widely available (historical) data on engineering systems, the data based approach became viable. In this approach, instead of developing a controlled experiment with the system, existing data is used to derive a relationship between inputs and outputs of the system [18].

Finally, when all the parts of the engineering system are well-understood, a theoretical model can be built to reflect the existing knowledge of the system [1]. This model-based approach combines all the parts of the system to explain the relationship between the inputs and outputs of the system. Interventions can be evaluated by testing the response of the model to changes in input data and the parts of the model.

These approaches are not mutually exclusive and are often combined to grasp the full complexity engineering systems. Each of these evaluation methods has been adopted across fields, both in academia and industry. Note that all approaches can be used to study interventions both qualitatively and quantitatively. The measurement scale depends on the type of intervention and the desired analyses that support the interpretation of the results. This section continues to discuss each of the evaluation methods.

A. Experimental Approach

The experimental approach for the evaluation of engineering system interventions relies on the comparison of two sets of empirical results before and after the intervention. The main benefit of this approach is that the results are real. As long as the measurement process is kept constant for both measurements, the empirical results reflect a change in the objects of the empirical relational system (or real world) [19]. Furthermore, the results from such an experimental approach hold for both behavioral and structural interventions. Note that experiments are also valuable to study specific pieces of engineering systems with small scale experiments, often in a well-controlled environment.

The experimental approach, however, has numerous disadvantages. Engineering systems are generally large, critical systems intertwined with the daily routine of the population [20]. Experimenting with these systems to find out which approach works best, potentially rebuilding systems multiple times, is a tremendous waste of money [17]. Furthermore, the execution of such an experiment is time consuming, and potentially reckless. The experimental approach should, therefore, only be used sparingly and mainly to inform the planning of future interventions (e.g. as in the case of pilot-projects) [21]. The value to provide "lessons learned" to future interventions should not be overestimated. Another downside of the experimental approach is that it is a black box model. The system as a whole is overhauled, but it may be unclear how external factors have changed between the time of the baseline measurement and the post-implementation measurement.

B. Data Driven Approach

The data driven approach to the evaluation of engineering system interventions relies on the definition of a statistics-based formal relational model between inputs and outputs. This model can be used to evaluate a behavioral intervention by estimating the response of the system to changing inputs. Generally, six types of data analysis are distinguished [22]:

Descriptive data analyses aim to describe the data without interpretation [23]. The most commonly used statistics in quantitative descriptive analyses are the sample mean and the sample standard deviation. A summary statistic for nominal measurements is a frequency analysis.

Exploratory data analysis provide a description and interpretation of the data aimed at providing insight into a problem [24]. The goal of exploratory data analysis is to find the "story" of the data, detect patterns and trends, and inform deeper study of the data. Some of the most common techniques include graphical representation of the data with boxplots, dotplots, or kernel density functions. Exploratory data analysis can also include preliminary model building and subset analyses.

Inferential data analysis aims to provide general facts about a certain type of systems given a limited amount of data [25]. It quantifies the correlation between measurements to provide insight in the generalizability of the patterns in the data. The two major branches in inferential data analysis are estimation and hypothesis testing. The former contains the main methods of point estimation and interval estimation. The latter contains a wide range of tests appropriate for different types of analyses. A non-exhaustive list of hypothesis tests is provided below [26]: 1) t-Test for independent means, 2) t-Test for Correlation Coefficients, 3) One-way ANOVA, 4) Analysis of Covariance, 5) Two-way ANOVA, 6) One-way repeated Measures ANOVA, 7) t-Test for Regression Coefficients, 8) Chi-Square for Contingency Tables.

Predictive data analysis measurements of a subset to predict the measurement on a single person or unit. The algorithms in this field are evolving quickly and they are often classified into supervised learning, and unsupervised learning. Supervised learning aims to learn a function that couples inputs to outputs from data that contains both inputs and outputs. A non-exhaustive list of supervised learning algorithms is [27]: 1) Support Vector Machines, 2) Neural nets, 3) Logistic Regression, 4) Naive bayes, 5) Memory-based learning, 6) Random forests, 7) decision trees, 8) bagged trees, 9) boosted stumps. Unsupervised learning is predictive data analysis without a pre-identified output or feedback. Some typical unsupervised learning examples are [28]: 1) Clustering, 2) Association rules, 3) Self-organizing maps.

The final two methods, **Causal data analysis** and **Mechanistic data analysis** rely on a theoretical understanding of the measured system and are used in conjunction with model-based evaluation approaches. Causal data analysis derives an average effect of one measurement on another, whereas Mechanistic data analysis aims to determine the relationship between two measurements under all conditions.

All analyses can be used to inform the design of the intervention. However, for the definition of the formal relational model that "predicts" the relationship between inputs and outputs after the intervention, only the last three types are appropriate. Note that the statistical model requires data beyond historical data of the original system [18]; for example from other systems comparable to the post-intervention system.

The benefits of the data driven approach are that it is both cheap and quick. The cost of collecting and storing data has plummeted while the availability has soared. In combination with rapidly evolving computational resources that can analyze the data, the creation of a data based model has become very affordable. Furthermore, the rise of cloud computing enable extremely fast analysis of the data.

The downsides of the data driven approach are related to the fact that statistical models are a black box [18], [29]. As a result, it is impossible to truly understand the elemental dynamics that define the overall system behavior. This is especially true for more advanced and automated statistical models based on neural networks and deep learning [30]. As a result of the opaque nature of the model, the study of structural interventions is not possible. The model loses its generalizability when the basic equations (or assumptions) are changed. Finally, the data driven models rely on the assumption that the system is stationary. In order to analyze interventions that break the "business-as-usual" case, data based approaches to intervention evaluation are insufficient.

In conclusion, data driven models are predominantly appropriate to analyze behavioral interventions in systems where the "mechanistic" science is not fundamentally understood. However, the analysis of structural interventions, or interventions that break the assumption of "business-as-usual" in any way, cannot be performed with data driven evaluation approaches.

C. Model-Based Approach

The model-based approach to the evaluation of engineering system interventions relies on the construction of a formal relational system based on knowledge of the empirical system [31]. The formal relational system is constructed to represent the dynamics of each of the elements in the empirical relational system. The combination of each of the elemental models create full system results that match the observed numerical results as derived from the measurement of the real-world system. The intervention is evaluated by implementing new or changed elemental models in the formal relational system. The empirical results interpret the numerical results of the two formal relational systems. Section V provides a closer look at the different model-based approaches to evaluating engineering system interventions.

The main benefit of the model-based approach is its transparency [32]. The elements in the models are known and have individual properties. The properties may include first principle-based dynamics. Furthermore, the model-based approach supports the evaluation of both behavioral and structural interventions. The model elements may be adjusted in their behavior, or be changed all-together.

The main downside of the model-based approach is that a deep knowledge of the engineering system is required to build a model that matches the real-world measurements [32].

In conclusion, model-based intervention evaluation is specifically valuable when used to represent a system that is well-known. It provides a transparent approach to the evaluation of both structural and behavioral interventions. In recent years, a discussion around the "end of theory" has emerged. The chapter addresses this discussion explicitly in the next section (Section V), together with an in-depth discussion of the model-based intervention evaluation methods. This section discussed three central approaches to the evaluation of engineering system interventions with their respective strengths and weaknesses. The next section provides extra detail on the last of these approaches.

V. Model-Based Intervention Evaluation

The previous section provided a comparison of the different methods for the evaluation of engineering system interventions. This section takes a closer look at the model-based intervention evaluation methods. Some literature has posited the "end-of-theory" given the explosion in the availability of data [33]. This section, however, demonstrates that theory plays an essential role in the future of engineering systems [34]–[36]. The discussion is structured in congruence with the classification of modeling methods as displayed in Figure 3.

The development of theory is critical to the future of engineering system design and intervention evaluation because ...

- ... it defines meta-data features in data collection.
- ... it ensures a deep understanding of the modeled system so that both structural and behavioral interventions are understood.

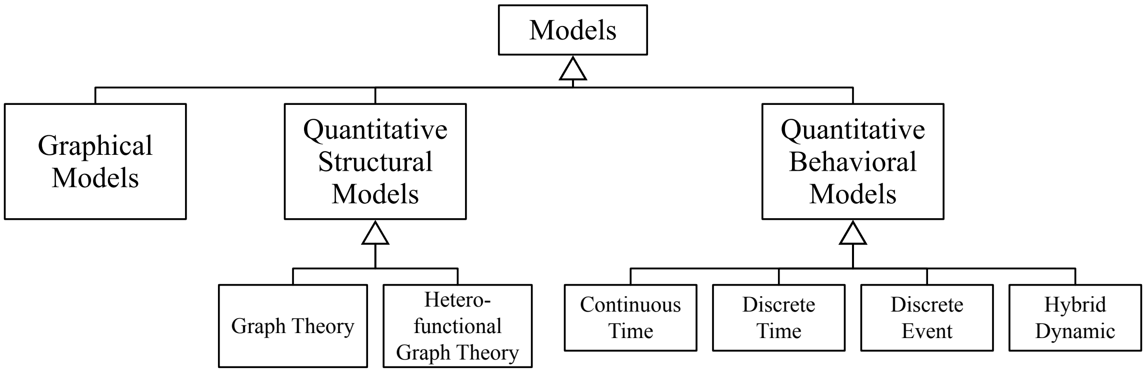


Fig. 3. Classification of modeling methods for the evaluation of engineering system interventions.

- ... it ensures a deep understanding of the modeled system such that the knowledge gaps are explicit. It requires assumptions and has the ability to inform future research (to test those assumptions).

Model-based evaluation of interventions do not forego the use of data and experiments. Rather, they leverage those in testing assumptions and creating a deeper understanding by extensive simulation and testing.

A. Graphical Models

The first class are graphical models. These models have been used to describe a wide range of systems, from technical to socio-economic[1]. Graphical models are qualitative in nature and they are often used to communicate the structure of a system. Furthermore, they are also used to communicate qualitative information and the ontology of a system or a class of systems. Furthermore, graphical models are not limited in the heterogeneity of the modeled system.

The downside of graphical models is the lack of support for quantitative analyses of the models. However, some methods have been developed to gain quantitative insights based on graphical modeling methods. These are often developed as part of a specific software package for the modeling method.

Below, a number of graphical modeling methods are introduced as a rough overview of the landscape. This list is not exhaustive but it provides the reader with a starting point.

IDEF0 diagrams enable the decomposition and architecture of system function [37]. For each function, IDEF0 lays out the inputs, controls, and mechanisms required to create the output. For clarity, the method relies on aggregation and decomposition of processes to limit the number of processes to six per layer of modeling abstraction. IDEF0 is one of the IDEF family of modeling languages. These languages have been developed starting in the 1970's with funding from the U.S. Air Force.

[1]Note that this definition is distinct from the graphical models in the field of machine learning in which graphical models refer to "graph-based" models, as described in Section V-B.

Unified Modeling Language (UML) was developed to provide a consolidated approach to object-oriented modeling methods [38]. UML was originally intended for software and firmware, but its strengths were recognized and the methods were applied to other fields.

Systems Modeling Language (SysML) borrows many features of UML and customizes them for cyber-physical systems. These include block definition diagrams and activity diagrams. SysML also includes a new set of diagrams to address the physical nature of these systems (e.g. the internal block definition diagram) and direct support for requirements engineering [16]. SysML is the most commonly used modeling language among systems engineers.

Model-based Systems Engineering created the Systems Modeling Language (SysML) as an abstracted graphical model with sufficient ontological breadth to integrate and synchronize more detailed domain-specific engineering models. SysML is not meant to develop complex mathematical models that provide engineering insight, as it is qualitative and graphical in nature. Rather, SysML provides systems engineers and project managers with a tool by which to quickly understand the overall structure and behavior of a system and its component modules so as to coordinate its engineering development in large and often multiple engineering organizations.

SysML leverages multiple modeling frameworks to represent the full breadth and complexity of an engineering system. This multitude of diagrams allows the modeler to separate, for example, form from function to study the processes in a solution neutral environment. The downside of using SysML is that the modeler needs to leverage the right diagrams to model the system.

Object-Process Methodology (OPM) has been developed explicitly for the modeling of general purpose systems with both system form and behavior in mind [39]. OPM describes form and function in a single diagram, with a single, consistent hierarchy. OPM has the benefit of having a single hierarchical model, and using a single type of diagram to represent the full system. However, OPM is missing the breadth to capture all aspects of a system.

Business Process Model and Notation (BPMN) is developed to support decision making around business processes [40]. The goal is to provide a language that can be intuitively understood by all stakeholders of the process. BPMN has overlap in functionality with activity diagrams in SysML, but BPMN is specifically designed for business processes and activity diagrams have a much broader applicability.

Causal-Loop Diagrams have been used to describe socio-technical systems. These use a directed graph approach to connect (hard and soft) variables as feedback loops. Causal-loop diagrams are easy to understand by stakeholders and can enable conversations about the dynamics of a system. The downside is that causal-loop diagrams quickly become complex and that the method doesn't lend itself for a hierarchical decomposition of the system. *"System Dynamics"* is a quantification of causal-loop diagrams. It was first developed in the '50s at MIT to model nonlinear behavior with stocks, flows, and feedback loops [41]. Over time, it has evolved to address a variety of dynamically complex systems. System Dynamics can be used both qualitatively, to describe and model systems, or quantitatively, to simulate dynamic behavior with the VenSim or Stella software packages.

B. Quantitative Structural Models

Quantitative structural models mathematically describe a systems structure.

Definition 10. System Structure [12], [32] is defined by the parts of a system and the relationships amongst them. It is described in terms of 1.) the system boundary, 2.) the formal elements of the system 3.) the connections between them 4.) the functional elements of the system and 5.) their allocation to the formal elements.

Quantitative structural models have been used extensively to describe both social and technical systems. In all cases, they rely heavily on graph theoretical concepts.

1) Graph Theory:

A **Network** (or graph $\mathcal{G}$) is a general means of representing patterns of connections or interactions between parts of a system [42]. The parts of the system are represented as nodes (or vertices $\mathcal{V}$). The connections or interactions are represented as lines (or edges $\mathcal{E}$). In addition to this set-theoretic definition, graph theory provides incidence and adjacency matrices as means of algebraic analysis. Networks are used to study systems in a wide variety of disciplines including the Internet, power grids, transportation networks, social networks, citation networks, biochemical networks, and neural networks among others. Objectively speaking, the definition of a graph $G = \{V, E\}$ captures only the first three (of five) parts of system structure. Consequently, one of the major shortcomings of Graph Theory is the failure to represent heterogeneity in networks as a result of the simplicity of its mathematical structure. Instead, many works attribute additional data *features* to graphs to expand their utility.

The **Design Structure Matrix**, for example, is a type of network modeling tool [43] that seeks to distinguish the different types of interconnections within a system. The four types of Design Structure Matrix models are 1) product architecture, 2) organization architecture, 3) process architecture, and 4) multidomain architecture.

Multilayer Networks expand on existing network theory to accommodate the study of networks with heterogeneity and multiple types of connections [44]. Over the past decade, numerous methods have tried to provide a consistent approach to model these networks-of-networks. However, as discussed by Kivela et al, all these multilayer network methods have their respective modeling limitations.

2) Hetero-functional Graph Theory:

Hetero-functional graph theory has emerged over the past decade to be the first quantitative structural model that captures all five parts of system structure [32]. It enables the structural modeling of a heterogeneous large flexible engineering system and explicitly accommodates all five types of system processes (i.e. Transform, Transport, Store, Exchange, and Control) and all five types of operands (i.e. Living Organisms, Matter, Energy, Information, and Money) that regularly appear in engineering systems [1]. Furthermore, Hetero-functional Graph Theory has been used as the underlying structure for dynamic system models across many different application domains including power, water, transportation, production, and healthcare systems. It has also been used to study the interdependencies of these systems within the context of interdependent smart city infrastructures.

C. Quantitative Behavioral Models

Quantitative behavior models can be broadly classified as 1) Continuous Time Behavioral Models, 2) Discrete Time Behavioral Models, 3) Discrete Event Behavioral Models, and 4) Hybrid Dynamic Behavioral Models.

1) Continuous Time Behavioral Models:

Continuous-Time and Discrete-Time Behavioral Models are closely related and can both be further classified into time-varying vs. time-invariant and linear vs. non-linear models. For more detail about that decomposition the authors refer the reader to the first chapter in [45].

Systems of Ordinary and Partial Differential Algebraic Equations (ODEs, PDEs & DAEs) are used to describe continuous time behavioral models. ODEs are often used to describe "lumped" systems while PDEs are used to describe distributed behavior (e.g. the traffic density along a stretch of road). Because it is often analytically or computationally intractable to use a truly distributed PDE, systems of ODEs arranged in a graph structure are often used instead. Bond graphs and linear graphs, for example, are well-known techniques that superimpose the constitutive laws of engineering physics onto the structure of a physical engineered system. Furthermore, pseudo-steady-state assumptions are often made so that a subset of the differential equations are effectively replaced by algebraic equations to form differential algebraic equations as described in Section II. Several software packages have been developed to simulate the systems of DAEs. These include Simscape by Matlab, OpenModelica and Dymola based on the Modelica language.

Agent-Based Modeling (ABM) goes beyond the dynamic laws of engineering physics to study socio-technical and socio-economic systems. ABM leverages dynamic interactions between autonomous entities called agents [46]. As the agents interact with each other, their individual processes and functions result in an emergent system behavior. This "bottoms-up" approach to modeling results in a number of benefits. ABM has the ability to predict emergent phenomena that often defy normal intuition. Furthermore, ABM provides a natural description of a system, especially for socio-technical systems in which individuals make decisions about their use of technical systems. Finally, ABM is flexible in that it can be expanded for the number of entities and their interactions. It also allows for changing levels of aggregation of agents in agent-groups.

2) Discrete Time Behavioral Models:
In contrast to the continuous time models, discrete time models are based on sampled data points or signals in digital form [47]. The rise of digital information technology has increased the need for a deep understanding of discrete time behavior and the corresponding mathematics.

Models of engineering systems can be developed from theory using either continuous or discrete mathematics. However, whenever data is collected, discrete-time models are the natural first choice. In either case, both types of models can be readily transformed from one to the other. In the case of linear systems, discrete-time systems of equations can be solved algebraically with the use of the Z-transform in much the same way that continuous-time systems can be solved algebraically with the Laplace transform.

The decision to use either continuous or discrete mathematics to model an engineering system depends primarily on the role of data and its discretization. In many cases the data is intrinsically discretized, or the data-collector has made pseudo-steady-state assumptions that force discrete-time step-wise evolution of algebraic equations. In other cases, data is not available and so idealized differential equations can be used. Finally, digital systems are more accurately represented with discrete time models, and engineering physics are generally more accurately represented with continuous-time models.

3) Discrete-Event Behavioral Models:
Discrete event behavioral models move from a time-driven view of the world to one that is *event-triggered*. In such a case, the system remains in a discrete state until such a moment where an event causes the system to flip into another state. Many discrete-event engineering systems exist; particularly as a result of automation where the underlying code is itself event-driven. Furthermore, discrete-event models always have discrete-state that is usually denoted by integers (rather than real or complex numbers).

Automata are one type of discrete-event model that are defined by a finite and countable set of discrete-states that each represent some phenomenon (that is often qualitative in nature). This includes on/off states as well as hot/cold or red/yellow/green. These states are described by nodes. Meanwhile, arcs are used to describe the event triggers that allow a switching behavior from one state to another. These triggers can be either endogenous or exogeneous rules but are often described by Boolean expressions (i.e. if $x \geq 0$ then switch from State 1 to State 2). While Automata have deep roots in theoretical computer science, they have since found broad application in describing the operational behavior of many engineering systems that have an underlying discrete decision-space. Automata are also often useful to describe operational modes of systems (e.g. normal, emergency, and restore) [45]. Despite these many strengths, the primary weakness of automata is that they have a centralized notion of state; and consequently all the states must first be enumerated in order for the complete automata to be well-defined.

Markov Models are a type of stochastic automata. They have been used to describe decision-making processes in a dynamic and stochastic environment [48]. Markov models have one of a finite number of states and stochastic events causes transitions between states. The evolution of state is tracked with each passing event or decision. Markov Chains are a type of Markov model in which the probabilities of transitions are fixed over time. These Markov models can be used to support decision-making in that they can help to estimate the effects of a certain decision, including subsequent decisions of others actors in the system.

Petri nets are another type of (deterministic) discrete-event model. Unlike automata, they have a decentralized description of state. In their simplest form, Petri nets consist of a set of places that define a state space, transitions that define events between a given pair of places, and a set of directed arcs that connect places and transitions [45]. In effect, these arcs create a *bipartite graph* between the sets of arcs and events. Furthermore, tokens are stored in places and are moved as each transition is "fired". The state of the system as a whole is described by a vector showing the number of tokens in each place. While Petri nets and automata have equal modeling power in that one can be mathematically transformed from the other (without loss), Petri nets can describe a relatively large number of automaton states with a relatively small number of places. Furthermore, because Petri nets are often represented graphically they often lends themselves to modeling distributed engineering systems such as warehouses, manufacturing systems, or supply chains more generally. Finally, in recent decades, the Petri net literature has expanded to accommodate time-driven dynamics through Timed and Time Petri

Nets. They have also incorporated various types of stochasticity with stochastic and fuzzy Petri nets.

4) Hybrid Dynamic Behavioral Models:
Hybrid dynamic behavior models combine the attributes of continuous/discrete time models with discrete-event models [49]. Generally speaking, they consist of a top "layer" described by either an automata or Petri net whose dynamics are either deterministic or stochastic. The bottom layer has a system of differential algebraic equations for each discrete state defined in the top layer. A classic example is the thermostat in a house. When the temperature is above a specified threshold, the heating system is idle. However, as soon as the temperature drops below the threshold, the heating system is activated and starts to heat the house. The model that is used to describe the "idling" state is distinct from the model that describes the "heating" state of the system.

Although hybrid dynamic systems have tremendous relevance to the understanding of engineering systems and their interventions, they remain at the cutting-edge of systems research. First, hybrid dynamic models often rely on discipline-specific DAE models. Consequently, some researchers resort to strapping together multiple (often off-the-shelf) simulators within co-simulation environments. In other cases, researchers develop custom simulators in order to address the specific needs of the engineering system under study. The literature contains many such simulators [50]. Finally, from an analytical perspective, there is a severe lack of theory that combines both discrete and continuous states. Consequently, many of the typical analytical methods applied to continuous-time systems (e.g. stability theory) or discrete-event systems (e.g. reachability analysis) can not be readily applied to hybrid dynamic systems.

VI. Conclusion and Future Work

This chapter has provided an thorough overview of engineering system interventions and their evaluation in an application-neutral language. It distinguished between interventions that change system behavior and those that change system structure. The type intervention dictates the type of evaluation and measurement that can be applied; be it experimental, data-driven, or model driven. The chapter was brought to a close with a taxonomy of engineering system models including graphical, quantitative structural, and quantitative behavioral models.

In regards to the last of these, hybrid dynamic models, while complex, have the greatest applicability to the growing complexity of today's engineering systems. There is a profound need to develop engineering system models that capture both its continuous-time as well as discrete event dynamics for the simple reason that engineering systems are continually changing structure while also evolving their system behavior. Such models grow our ability to practically study these engineering systems from a simulation perspective. That said, concerted theoretical effort can serve to provide deep analytical and generic insights into the structural, behavioral, and life-cycle properties of these systems.

The recent COVID-19 pandemic has highlighted the need for a deep understanding of both the technical and the social side of engineering systems. The engineering aspects of the dynamics of the pandemic are well-understood. For example, the global transportation system has enabled the virus to spread rapidly over the globe. However, the impact of human behavior as part of social interactions is still unclear. Based on experiments and data collection, scientists have tempted to infer how the virus is most likely to infect other humans. The interaction between the well-understood technical side of the pandemic and the poorly understood social (human) side can be accurately represented by a hybrid dynamic model.

In addition to hybrid dynamic models, hetero-functional graph theory provides an avenue to investigate the ***complete*** structure of an engineering system. Such an approach does not require the extensive effort that is often needed to develop simulations of hybrid dynamic systems. Instead, UML/SysML models can be straightforwardly developed and then instantiated and translated automatically to produce hetero-functional graphs. In recent years, the network science community has provided an explosion of computational results over (traditional) graphs. There is great potential to apply similar approaches to hetero-functional graphs and capture the true heterogeneity found in modern engineering systems.

References

[1] O. L. De Weck, D. Roos, and C. L. Magee, *Engineering systems: meeting human needs in a complex technological world*. Cambridge, Mass.: MIT Press, 2011. [Online]. Available: http://www.knovel.com/knovel2/Toc.jsp?BookID=4611http://mitpress-ebooks.mit.edu/product/engineering-systems

[2] J. Jacobs, *The death and life of great American cities*. Vintage, 1961.

[3] R. Lempert, "Agent-based modeling as organizational and public policy simulators," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 3, pp. 7195–7196, 2002.

[4] K. Yeo, "Planning and learning in major infrastructure development: systems perspectives," *International Journal of Project Management*, vol. 13, no. 5, pp. 287–293, 1995.

[5] S. O. Muhanji, C. Below, T. Montgomery, and A. M. Farid, "Enabling a Shared Integrated Grid in New England's Energy Water Nexus," in *2019 IEEE International Symposium on Technology and Society*, Boston, MA, USA, 2019.

[6] C. H. Edwards, D. E. Penney, and D. Calvis, *Differential equations and linear algebra*. Pearson, 2018.

[7] B. T. Kulakowski, J. F. Gardner, and J. L. Shearer, *Dynamic modeling and control of engineering systems*. Cambridge University Press, 2007.

[8] A. M. Farid, "An engineering systems introduction to axiomatic design," in *Axiomatic Design in Large Systems: Complex Products, Buildings & Manufacturing Systems*, A. M. Farid and N. P. Suh, Eds. Berlin, Heidelberg: Springer, 2016, ch. 1, pp. 1–47. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-32388-6

[9] M.-W. Dictionary, "Merriam-webster," 2019. [Online]. Available: https://www.merriam-webster.com/dictionary/

[10] D. Rowell and D. N. Wormley, *System dynamics: an introduction*. Upper Saddle River, NJ: Prentice Hall, 1997.

[11] D. M. Buede, *The engineering design of systems: models and methods*, 2nd ed. Hoboken, N.J.: John Wiley & Sons, 2009.

[12] A. M. Farid, "Reconfigurability Measurement in Automated Manufacturing Systems," Ph.D. Dissertation, University of Cambridge Engineering Department Institute for Manufacturing, 2007. [Online]. Available: http://engineering.dartmouth.edu/liines/resources/Theses/IEM-TP00.pdf

[13] L. Finkelstein, "Theory and philosophy of measurement," *Handbook of measurement science*, vol. 1, pp. 1–30, 1982.

[14] ——, "Problems of measurement in soft systems," *Measurement*, vol. 38, no. 4, pp. 267–274, 2005.

[15] E. Čech and M. Katětov, "General metric spaces," *Point sets*, 1969.

[16] SE Handbook Working Group, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. International Council on Systems Engineering (INCOSE), 2015.

[17] A. J. Felson and S. T. Pickett, "Designed experiments: new approaches to studying urban ecosystems," *Frontiers in Ecology and the Environment*, vol. 3, no. 10, pp. 549–556, 2005.

[18] S. Washington, M. G. Karlaftis, F. Mannering, and P. Anastasopoulos, *Statistical and econometric methods for transportation data analysis*. CRC press, 2003.

[19] V. C. Broto and H. Bulkeley, "A survey of urban climate change experiments in 100 cities," *Global environmental change*, vol. 23, no. 1, pp. 92–102, 2013.

[20] A. Karvonen and B. Van Heur, "Urban laboratories: Experiments in reworking cities," *International Journal of Urban and Regional Research*, vol. 38, no. 2, pp. 379–392, 2014.

[21] F. Caprotti and R. Cowley, "Interrogating urban experiments," *Urban Geography*, vol. 38, no. 9, pp. 1441–1450, 2017.

[22] J. T. Leek and R. D. Peng, "What is the question?" *Science*, vol. 347, no. 6228, pp. 1314–1315, 2015.

[23] W. C. Navidi, *Statistics for engineers and scientists*. McGraw-Hill Higher Education New York, NY, USA, 2008.

[24] J. T. Behrens, "Principles and procedures of exploratory data analysis." *Psychological Methods*, vol. 2, no. 2, p. 131, 1997.

[25] R. Lowry, *Concepts and applications of inferential statistics*. Online, 2014. [Online]. Available: http://vassarstats.net/textbook/index.html

[26] L. B. Christensen, B. Johnson, L. A. Turner, and L. B. Christensen, *Research methods, design, and analysis*. Pearson New York, 2011.

[27] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 161–168.

[28] C. Sammut and G. I. Webb, *Encyclopedia of machine learning and data mining*. Springer Publishing Company, Incorporated, 2017.

[29] C. Caminha, V. Furtado, V. Pinheiro, and C. Silva, "Micro-interventions in urban transportation from pattern discovery on the flow of passengers and on the bus network," in *2016 IEEE International Smart Cities Conference (ISC2)*. IEEE, 2016, pp. 1–6.

[30] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *arXiv preprint arXiv:1708.08296*, 2017.

[31] G. Guizzardi, "On ontology, ontologies, conceptualizations, modeling languages, and (meta) models," *Frontiers in artificial intelligence and applications*, vol. 155, p. 18, 2007.

[32] W. C. Schoonenberg, I. S. Khayal, and A. M. Farid, *A Hetero-functional Graph Theory for Modeling Interdependent Smart City Infrastructure*. Berlin, Heidelberg: Springer, 2018. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-99301-0

[33] C. Anderson, "The end of theory: The data deluge makes the scientific method obsolete," *Wired magazine*, vol. 16, no. 7, pp. 16–07, 2008.

[34] P. Smaldino, "Better methods can't make up for mediocre theory," *Nature*, vol. 575, no. 7781, pp. 9–9, Nov 2019. [Online]. Available: http://dx.doi.org/10.1038/d41586-019-03350-5

[35] F. Mazzocchi, "Could big data be the end of theory in science? a few remarks on the epistemology of data-driven science," *EMBO reports*, vol. 16, no. 10, pp. 1250–1255, 2015.

[36] S. Succi and P. V. Coveney, "Big data: the end of the scientific method?" *Philosophical Transactions of the Royal Society A*, vol. 377, no. 2142, p. 20180145, 2019.

[37] Anonymous, *Systems Engineering Fundamentals*. Fort Belvoir, Virginia, USA: Defense Acquisition University Press, 2001, no. January.

[38] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*. Reading, Mass.: Addison-Wesley, 2005.

[39] D. Dori, *Object-process methodology : a holistics systems paradigm*. Berlin ; New York: Springer, 2002.

[40] OMG, "Notation (bpmn) version 2.0," Object Management Group, Tech. Rep., 2011.

[41] J. W. Forrester, "System dynamics, systems thinking, and soft or," *System dynamics review*, vol. 10, no. 2-3, pp. 245–256, 1994.

[42] M. Newman, *Networks: An Introduction*. Oxford, United Kingdom: Oxford University Press, 2009. [Online]. Available: http://books.google.ae/books?id=LrFaU4XCsUoC

[43] S. D. Eppinger and T. R. Browning, *Design structure matrix methods and applications*. Cambridge, Mass.: MIT Press, 2012.

[44] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of complex networks*, vol. 2, no. 3, pp. 203–271, 2014.

[45] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. New York, NY, USA: Springer, 2007.

[46] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the national academy of sciences*, vol. 99, no. suppl 3, pp. 7280–7287, 2002.

[47] K. Ogata, *Discrete-time control systems*, 2nd ed. Englewood Cliffs, N.J.: Prentice Hall, 1994.

[48] F. A. Sonnenberg and J. R. Beck, "Markov models in medical decision making: a practical guide," *Medical decision making*, vol. 13, no. 4, pp. 322–338, 1993.

[49] A. J. Van Der Schaft and J. M. Schumacher, *An introduction to hybrid dynamical systems*. Springer London, 2000, vol. 251.

[50] D. F. Allan and A. M. Farid, "A benchmark analysis of open source transportation-electrification simulation tools," in *2015 IEEE Conference on Intelligent Transportation Systems*, Las Palmas de Gran Canaria Canary Islands, Spain, 2015, pp. 1–7. [Online]. Available: http://dx.doi.org/10.1109/ITSC.2015.198