Engineering Informatics to Support Civil Systems Engineering Practice

Timo Hartmann(✉)

Civil Systems Engineering, TU Berlin, Berlin, Germany
timo.hartmann@tu-berlin.de

Abstract. Systems engineering is the interdisciplinary engineering field that focuses on the design of complex physical systems to optimize the system's performance over its life-cycle. To support such optimization efforts a number of computational modeling methods are required: ontological modeling, stochastic modeling, and process simulation modeling. Despite this need, the field of systems engineering has mainly focused on the development and discussion of managerial methods. This paper tries to provide a first starting point for a discussion about a framework to understand how the above mentioned computational methods can support system engineers. The paper introduces a first set of important methods and tries to integrate them in an overall framework for analysing engineered systems from different points of view. For each of the methods we also provide a simple illustrative example from our ongoing systems engineering teaching efforts at the TU Berlin.

1 Introduction

The basic idea behind systems engineering is that a high performing product can only be designed if each of the product's components and physical subsystems work in an integrated way together. An important aspect to allow engineers to understand the level of integration of a system is the modeling and simulation of a system to understand how well different design options of the system perform with respect to a set of previously defined requirements. These aspects of modeling and simulation have not yet been widely discussed in the scientific discourse. The discourse in the field of systems engineering is still mainly focused on the aspects of requirements engineering, as well as, the abstract modeling of system components and interfaces.

To start a discussion about the required computational methods to support systems engineering efforts, this position paper sets out to provide an overview of three important areas of advanced computing: Product modeling, process modeling and data analytics. We argue that a focus on these three areas will allow the field to move towards the simulation based integrated practice that is at the core of systems engineering philosophy. Among these areas, product modeling forms the basis for the other two as it allows to understand the different components of a complex engineered system together with its characteristics and interfaces.

I. F. C. Smith and B. Domer (Eds.): EG-ICE 2018, LNCS 10864, pp. 262–275, 2018.
https://doi.org/10.1007/978-3-319-91638-5_14

Based on well defined product models different physical and social processes that influence some or all of a system's components, such as structural dynamics, thermal behavior, or traffic loads can be simulated. The product models allow for the integration of different process models and simulations that allows for a holistic understanding of the behavior of the overall system upon the influence of the different physical and social processes. At the same time, advanced data analytics allows us to understand the current and historical behavior of different system components and their interrelation to each other. Data analytics allows an alternative to the process modeling and simulation area for predicting a system's behavior in instances where historical data is available about the behavior of a system or of some of its components.

The paper will briefly introduce these three areas and their computational underpinnings. To this end, the paper will show how system engineers can apply these computational methods to gain a better understanding about a system to support engineering work. The computational methods will be illustrated using simple class room examples that we use in our teaching modules at the civil systems engineering department at the TU Berlin. I close the paper by the introduction of a theoretical framework that combines the three areas and that can be used to organize system engineering efforts.

2 Systems Thinking in Engineering

Every product engineers design and commission, be it a bridge, road, or building, is comprised of sub-components that stand in relation to each other. Through this relation, the different elements form a whole reacting to certain environmental influences, supporting civil life, from crossing rivers, to driving safely, from providing shelter to the outside world. In this sense, we can conceptualize each civil product as a system of elements that stand in relation to each other and thus form a whole that is more than the individual elements in isolation.

In traditional systems thinking (Luhmann 1984; Ropohl 2012), there are at least three important concepts of systems that can foster understanding about the basic composition of civil engineered products. First, systems need to be understood as functional in relation to their environment. For the concept system to make sense a clear distinction between the system and its environment needs to be present. A system can then be defined as a collection of elements that receive information or physical stimulus from their environment, internally process these information and stimula and provide some type of output or reaction. This view on systems is often labeled as functional. The functional concept of systems allows us to understand questions such as "What is this thing" or "What does this thing do", while it specifically does not look at the inner composition of the system (Ropohl 2009).

While the functional concept treats the internal composition of the system as a black box, the structural concept looks into the system. A system also needs to be understood as a set of individual elements that stand in relation to each other. Each of the elements can be connected to each of the other elements in different ways. How elements are connected to each other defines the inner structure of the system. Therefore, this second concept is often called the structural concept of systems.

Finally, each of the elements of the system can itself be considered as a system that stands in relation to an environment. At the same time, the environment of each system can also be conceptualized as a system itself. Any reflection about a system can hence comprise different levels of super- and sub-systems, something that is often referred to as systems of systems thinking (Luzeaux et al. 2013).

While the traditional field of systems engineering as a framework to guide the engineering process of complex technical systems has evolved independent of systems thinking, the three above concepts of systems can be mapped well to the different main tasks prescribed by the systems engineering methodology. Systems engineering has recently evolved as the leading management practice across all the engineering disciplines and prescribes a set of iterative processes to be applied for designing, developing, operating, and maintaining complex engineered products throughout their life-cycle (Kapurch 2010). Systems engineering focuses on optimizing an engineered product as a whole, balancing each of the required components of the entire product to achieve some given product requirements.

Models, as abstract representations of reality, of the engineering product can be seen at the core of the systems engineering approach. As modern engineered products are highly complex, abstract models are required to understand the behavior of the products and allow for optimizing the product's design. Through models complex engineered products are simplified and conceptualized as systems, consisting of interacting elements, that together have to react to environmental conditions. The models are then used to define different alternatives for connecting components and then testing these alternatives towards the requirements during the design process. This allows engineers to theoretically understand the behavior of their product before developing and operating the product in the physical world.

The three concepts of systems thinking introduced above are at the core of understanding how to abstract good models to support the three main processes of systems engineering: Requirements management, interface management, and iterative and hierarchical component engineering. For one, each model needs to allow for understanding whether a engineered product can fulfill specific requirements for its functionality. To allow engineers to manage the requirements well, functional system models are required that can be used to evaluate the behavior of a system according to different changing environmental conditions.

At the same time, system engineering is concerned with defining the functional structure of the engineered products in terms of the products' components and their relations. Models need to support engineers to understand which components are required, how the different components need to be related, and how the related components together can react to different environmental conditions. Additionally, from a life-cycle perspective, engineers need to already understand in early design phases how components can be exchanged, maintained, and recycled at different life-cycle stages of an engineered product. To support these tasks structural system models of the engineering product are required.

Finally, the systems engineering approach prescribes a highly iterative process during which each of the components of a system are split up in their smaller components. The NASA systems engineering handbook (Kapurch 2010) for example suggests that every component of a system should be divided in sub-components until the sub-component can be acquired on the market or ordered at a third party supplier. These components at the lowest level of the product hierarchy then should be assembled to sub-components, the sub-components are then again assembled to higher level components till a final product that fulfills the requirements is engineered. To support this product engineering approach hierarchical system models are required.

Without a doubt, the computer has become an indispensable tool to support all of the above modeling tasks during systems engineering. Despite the ubiquitous presence of computers in everyday engineering, there is little discussion about computational support within the systems engineering community. The next section, therefore summarizes three important computational methods for supporting the above described systems engineering process focusing on the three areas of ontological modeling, stochastic data analytics, and process modeling and simulation.

3 Fundamental Informatics to Support Civil Systems Engineering

3.1 Ontologies and Product Modeling

The computational discipline of ontology engineering is concerned with the formal naming and the definition of entities, their properties, and the relation between entities within a specific domain of discourse (Noy et al. 2001). In that sense the engineering of an adequate ontology describing a civil engineered product is at the core of any systems engineering effort. Without formally defining and naming the different elements and environmental influences of a system together with the different possibilities to relate the elements with each other and with the environment no computational possibilities to support the engineering effort would be possible. Figure 1 shows an illustrative example of an ontology that models the different elements of a bridge.

While an ontology is a conceptual formalization of the logic behind the elements of a system and their relations (Krötzsch et al. 2012), engineering is also always concerned with the physical embodiment of the system in the real world. How an engineered product is geometrically configured is an important step towards the realization of the physical product. The product configuration is also important during the simulated evaluation of the different alternatives for the final product. Based on an ontology describing an engineered product conceptually, geometrical parameters help to link the conceptual description to the geometrical description of the product. Parameters allow engineers to capture design knowledge and intent within flexible models that are automatically

Fig. 1. Example of an ontology that conceptually models the different components of a bridge.

updated when the defined parameters change (Geyer 2008). Parameters also allow to define product family and parts that describe sub-elements of the engineered system (Pahl and Beitz 2013). Finally, parameters can be steered by computational algorithms to quickly generate a large number of possible physical configurations of a system for the purpose of evaluating the designs (Flager et al. 2009).

Figure 2 shows an example of a parametric model that can generate different geometrical configurations of a bridge based on two input parameters - bridge length and transversal span. This parametric model can be used to generate a large number of different bridge alternatives varying in length and transversal spans. This generation can be computationally steered.

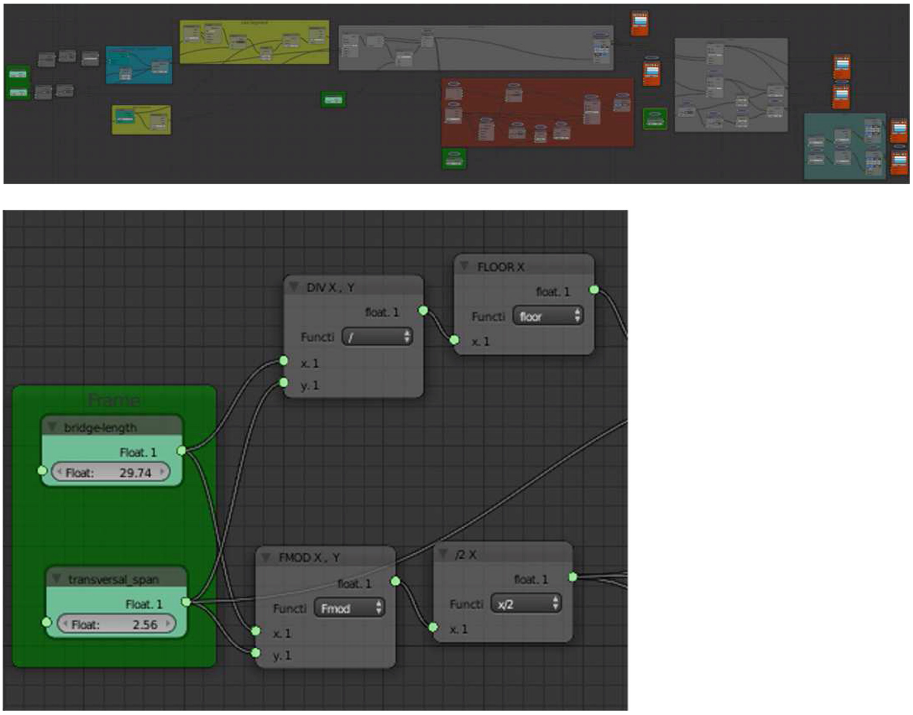

Fig. 2. Parametric model to generate the geometry of a bridge based on two input parameters. Upper part: The complete parametric model. Lower part: Detail of the model showing the two input parameters - bridge-length and transversal span

3.2 Data Analytics

Functional system models can only be established if the underlying relationship between the elements of a system are understood well. In cases such an understanding does not exist, the functional behavior of a system can still be represented with stochastic models (Matloff 2009).

Given a set of random observations about the stochastic behavior of a system or some environmental process that influences the system, statistical models can help to understand the random functional behavior of the system. Such an understanding, in turn, allows engineers to estimate the different possible conditions of the system behavior and how likely these conditions are. In particular, if an engineering effort is concerned with the safety of a system, a statistical model can help engineers to understand the magnitude of possible extreme conditions a system can be in to design systems that can withstand such conditions. Computationally engineers can for example use maximum likelihood methods to estimate the parameters for a given statistical model. To use maximum likelihood estimators, an engineer first assumes a possible joint density function for all observations. A more advanced method to estimate a probability density function for a given set of observations is the kernel density estimate. Kernel density estimation is a non-parametric method so that engineers do not need to make an a-priori assumption about an initial joint density function.

Next to the estimation of statistical models to estimate the stochastic behavior of a system or of an environmental process influencing the system, statistical testing methods can be applied to understand interaction effects between different elements of a system and different environmental influences of the system. A wide choice of statistical tests exists, but the two most commonly used tests are correlation tests and t-tests. Correlation tests allows to understand the connection of two elements based on a set of random observations of the behavior of the elements. The t-test on the other hand allows to compare two different groups of elements and can help engineers to understand whether two groups of systems or one system under two different environmental conditions behave significantly different.

To illustrate the above introduced basic statistical methods we use a data set collected for one of our earlier research studies (Ziari et al. 2016) with the aim to predict the deterioration behavior of roads according to different conditions based on a large data set of US highways collected by the Federal Highway Administration of the United States of America. The example is based on a very simple system model of the road, describing the road's physical composition as the thickness of the road's pavement layer and the thickness of its surface layer. As main performance measure for the quality of the road the roughness of the road's surface is used measured using the international roughness index. The road system is then influenced by a number of environmental processes related to the weather (annual average precipitation, annual average temperature, and annual average freeze index) and to traffic (annual average daily traffic, annual average daily truck traffic, single equivalent axle load). Observations for each of these system and environmental elements are collected from the freely available database of the Federal Highway Administration and the data set is described in detail in (Ziari et al. 2016).

As a start analyzing the data set, statistical models about the overall deterioration behavior of the road can be established using for example a maximum likelihood estimator. To understand the shape of the overall distribution the change in the roughness index of the roads after different years can be ploted as a histogram. Then suitable joint probabilitity density functions can be choses. For example, in this case, a gamma distribution seems to be a good choice as it will allow us to estimate parameters that seem to provide good estimates for the deterioration of the road across the different time spans. Now a computational maximum likelihood estimator can be used to define the parameters for each time span and provide us with joint probability density functions that can be used to understand the deterioration behavior of the road better. Figure 3 provides a visual summary of the above described example.

Beyond standard mathematical tests, advanced machine learning methods allow for developing detailed prediction models of the behavior of a functional system based on a set of observations about the behavior of a system's elements

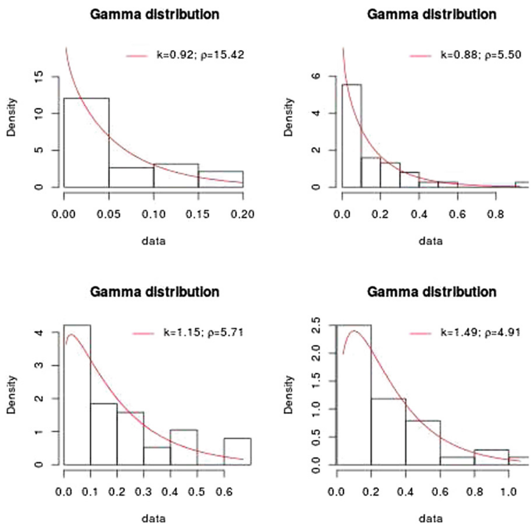

Fig. 3. Example of fitting a probability density function using the maximum likelihood estimator. Here we fit a gamma function to a data set that describes the deterioration of different road sections in the USA after one, two, three, and four years. The fitted function can then be used to sample values for stochastic simulations of road deteriorations

and its environment. Here the classical linear or non-linear regression methods require that an engineer defines a mathematical model of a system's structural behavior first. The classical regression methods will then estimate the parameters of the model based on a data set of previous observations. Advanced regression based methods, including support vector machines or artificial neural networks completely treat a functional system as a black box providing a mathematical model of the behavior of the system without any prior knowledge about the behavior of the system. A downside of these methods is, however, that they do not provide any new insights about the structural behavior of the system (Lantz 2013).

Bayesian and tree based machine learning methods are less accurate in their predictions of the functional behavior of a system. As an advantage, they provide insights into a system's structural behavior which often provides important insights for engineers (Lantz 2013). Finally, cluster based methods can provide categories for different behavioral states of a system grouped by environmental influences (Lantz 2013).

3.3 Process Modeling, Simulation, and Optimization

The above described statistical methods mainly help to understand the functional and, to a certain extent, the structural behavior of a system based on previously available observations of the system's behavior. In practice, often such observations are not available and therefore engineers have to rely on theoretical models that describe the structural behavior of a system. If implemented mathematically on a computer these models can be used to simulate the behavior of a system.

Traditionally, the core to many of these simulation methods are partial differential equations that model the change of some aspect of a system's behavior over time. Differential equations model the transition of a system's state assuming some underlying mathematical model involving the components of the system. Only simple linear partial differential equations can be solved analytical, so that computational solvers for partial differential equations have quickly become the norm within engineering practice (Farlow 1993).

Partial differential equations allow for modeling systems whose states are changing equally with each time step. Using functional programming techniques, however, also allow for modeling systems that change non-linearly. Such models are often referred to as discrete event methods and operate using an event queue that stores events that can be executed at arbitrary time steps. In particular, such discrete event simulation methods are valuable to model randomly occurring environmental events to provide a much deeper insight into a system's behavior than models that are based on partial differential equations only (Wainer and Mosterman 2016).

Simulation models based on partial differential equations can then be used to automate the analysis of the system models using the parametric modeling methods described earlier. Additionally, of course, variables describing environmental influences on the system can be varied in a similar manner. This allows for systematically changing the different initial input variables that either describe the behavior of the elements or the environment of the system. The range of all possible input variables is then often referred to as the parameter space. Combinatoric computational methods can be used to systematically analyse a large number of different alternatives within this parameter space. Such combinatoric analyses of the environmental factors modeled allow systems engineers to understand the behavior of a system under a large range of different outside factors. At the same time, different configurations of the parameters describing the system itself can be evaluated helping system engineers to develop optimal design configurations (Saltelli et al. 2000).

Each simulation result from the combination of different parameters can be considered as an observation in itself. Therefore, sets of simulation results can be analysed with the above described statistical methods. Such analysis allows for understanding relations between different parameters in the simulation models. This allows engineers to understand the importance of different parameters on the simulation outcome, a practice often referred to as sensitivity analysis (Saltelli et al. 2000). Parameters with little influence on the final outcome can be removed from the initial modeling equations and parameters with linear relations with each other can be represented as a single input factor (Forrester et al. 2008). Finally, machine learning methods can be used to train statistical prediction models. These prediction models can then be used instead of the partial differential based models which often allows to provide results in split seconds without running the often computationally expensive models (Forrester et al. 2008).

Often the parameter spaces are too large or the process models are too computationally expensive for simulating enough different parameter configurations so that system engineers can sufficiently understand how a system reacts to different environmental conditions and how a system can be ideally configured to cope with the different possible reactions. In these cases, computational sampling methods exist that can be applied to systematically search a vast design space using a well chosen amount of simulation runs (Saltelli et al. 2000). These sampling methods can be divided into purposeful samples that explore well chosen strata or in random samples that allow for an exploration according to the likelihood for the status of different environmental or system parameters. The stratified sampling methods help systems engineers to understand different well chosen groups of parameter configurations within different areas of the overall parameter space. This for example allows to include rare events that are of particular importance during safety engineering tasks. Random sampling methods, in turn, rather allow to understand system models with respect to their average and common behavior. To use statistical sampling methods a probabilistic distribution for environmental and system parameters needs to be assumed. This distribution can either be extracted from past observations using the above describe statistical methods or, alternatively, be assumed by specialists. Choosing sound distributions, in turn, allows to understand the general relation between elements of a system under general environmental conditions. Stratified sampling and statistical sampling can of course be combined in many different ways.

Finally, from a hierarchical system view, each of the sub-components of a system can itself be seen again as a functional system. In case enough observations about the behavior of such a functional system are available, each of the components can, in turn, be modeled using statistical computations techniques, in particular, using probability density modeling techniques, such as the maximum likelihood methods or kernel density estimates. Simulation models can than sample from the resulting probability density functions and the samples can serve as input for the theoretical structural system models. Environmental conditions can be modeled in the same manner. As samples from these probability density functions can provide arbitrary results, it is important to repeat the simulations multiple times sampling as many possibilities from the density functions as possible using so called Monte Carlo methods. The different simulation results, in turn, result in probabilistic distribution.

As an example, the above case of predicting road deterioration can be used again to illustrate the working of process model simulations in relation to sensitivity analysis and Monte Carlo methods. The example is based on a simple dynamic simulation model of two bridges that link two cities with each other. The model assumes that drivers tend to choose the bridge that is least deteriorated. At the same time, the more drivers travel over a bridge, the quicker the bridge deteriorates. We illustratively modeled these two processes for one of our

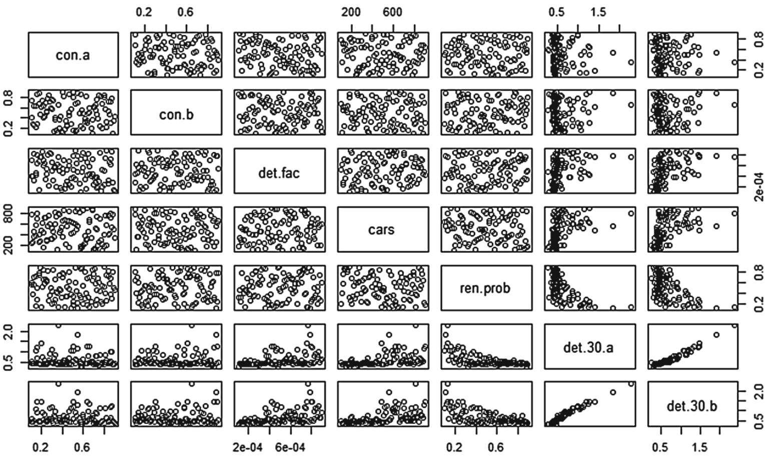

Fig. 4. Scatterplot resulting from a sensitivity analysis of a highly non-linear and stochastic model of road deterioration. The input variables of the initial road condition of two roads (con.a; con.b), the deterioration of the roads upon crossing of one car (det.fac), the number of crossing cars per year (cars), and the probability that a road is renovated in a year (ren.prob) are samples using the latin hypercube method. The output values road deterioration in 2030 of road a and road b (det.30.a, det.30.b) is simulated using a Monte Carlo approach. The scatterplot clearly shows that the initial road condition has little influence on the deterioration of the roads in 2013, while the deterioration factor and the number of cars have a positive influence, and the renovation probability a negative influence on the deterioration

teaching modules using two simple partial differential equations. Furthermore, the simple simulation model used stochastic distributed values the deterioration factor similar to the earlier introduced road example. To make the model highly non-linear we also assumed a certain probability that the bridge is renovated each year significantly reducing the existing road deterioration. We modeled this influence as a discrete event within the overall simulation model that can occur with a certain probability. A sensitivity analysis for this highly stochastic and non-linear model can for example sample the different parameter settings for each of the input values using a combined stratified and random sampling method (in this case the latin hypercube method) and based on the sampled parameter values can conduct a Monte Carlo simulation. A resulting scatter plot from this exercise is presented in Fig. 4. This scatterplot allows already to identify certain trends in the influence of the different input parameters on the bridge deterioration after a number of years.

4 Discussion

Figure 5 summarizes the presented computational methods within the concepts of the three different analytical views of systems. The behavior of functional

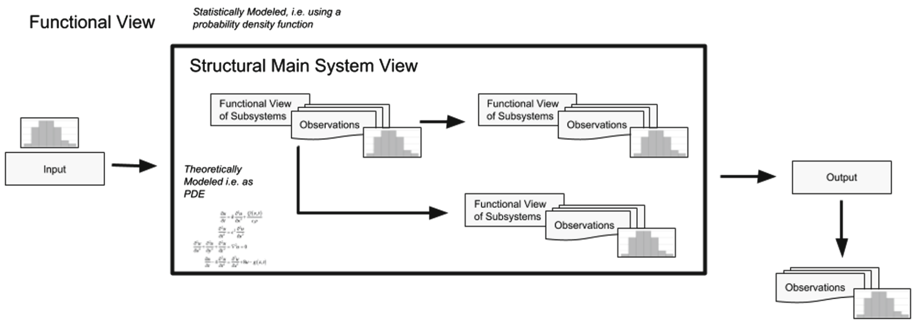

Fig. 5. Overall system based concepts of the paper

systems can be analyzed using statistical computational methods if there are a reasonable number of observations available about the input and behavior of a system. Systems can also be modeled using a functional view, by using partial differential equations and other computational simulation methods, such as discrete event simulations. These methods allow for the theoretical modeling of a system's behavior by describing the system's components and the relations between these components. The design of such functional simulation models requires an understanding of the system's components that can be gained through ontological modeling. Functional simulation models allow the evaluation of a large number of alternatives using parametric modeling methods that allow to systematically explore the combinatoric design space and the various environmental conditions a system might be subjected to. Finally, a hierarchical view of the system allows to combine statistical and theoretical modeling methods by the possibility to represent the behavior of selected components of a system through a stochastic functional model. In these cases, a system analysis based on simulation studies need to be subjected to Monte Carlo methods.

Table 1 summarizes the here presented computational methods that I consider as the basic methods within the overall toolkit of systems engineers. It is important to realize that these methods are just a suggestion of a set of basic methods and that I do not claim completeness. Other methods might be as relevant as the ones that are here presented. Candidates for further exploration would be for example optimization methods that allow to find optimal solutions within parametric search spaces. As quite some of the presented methods are computational very expensive methods to computationally parallelizise algorithms might also be important. Finally, the paper does not discuss any computational visualization methods. Nevertheless, I hope that this paper can provide a first starting point for future discussions about which of the methods are relevant and which are not.

The above summary of methods can provide academics with a good overview that can be used to design technically and computationally oriented systems engineering courses. The Bachelor and Master modules that we developed at the Technical University of Berlin can provide an example of how to integrate

Table 1. Overview of the introduced methods

Computational area	Computational method
Product system modeling	Ontology modeling
	Parametric modeling
Statistical system modeling	Maximum likelihood methods
	Kernel density estimates
	Correlation analysis
	t-Test
Theoretical system modeling	Partial differential equation solvers
	Discrete event simulation
	Sensitivity analysis
	Surrogate modeling
	Sampling methods
	Monte Carlo methods

the methods within a civil engineering curriculum. The summary can also help practitioners to understand how to support their practical system engineering efforts better using computational methods. In the end, if nothing more, we hope that the summary of computational methods together with my attempt to integrate the discussion into a system philosophical framework can help readers to better grasp the relevance of the introduced computational methods to support engineering tasks.

References

Farlow, S.J.: Partial differential equations for scientists and engineers. Courier Corporation, North Chelmsford (1993)

Flager, F., Welle, B., Bansal, P., Soremekun, G., Haymaker, J.: Multidisciplinary process integration and design optimization of a classroom building. J. Inf. Technol. Constr. (ITcon) **14**(38), 595–612 (2009)

Forrester, A., Keane, A., et al.: Engineering Design Via Surrogate Modelling: A Practical Guide. Wiley, Hoboken (2008)

Geyer, P.: Multidisciplinary grammars supporting design optimization of buildings. Res. Eng. Design **18**(4), 197–216 (2008)

Kapurch, S.J.: NASA Systems Engineering Handbook. Diane Publishing, Collingdale (2010)

Krötzsch, M., Simancik, F., Horrocks, I.: A description logic primer. arXiv preprint arXiv:1201.4089 (2012)

Lantz, B.: Machine Learning with R. Packt Publishing Ltd., Birmingham (2013)

Luhmann, N.: Soziale Systeme, vol. 478. Suhrkamp, Frankfurt am Main (1984)

Luzeaux, D., et al.: Systems of Systems. Wiley, Hoboken (2013)

Matloff, N.: From Algorithms to Z-scores: Probabilistic and Statistical Modeling in Computer Science. Creative Commons License (2009)

Noy, N.F., McGuinness, D.L., et al.: Ontology development 101: A guide to creating your first ontology (2001)

Pahl, G., Beitz, W.: Engineering Design: A Systematic Approach. Springer Science & Business Media, London (2013)

Ropohl, G.: Allgemeine Technologie: eine Systemtheorie der Technik. KIT Scientific Publishing, Karlsruhe (2009)

Ropohl, G.: Allgemeine Systemtheorie: Einführung in transdisziplinäres Denken. edition sigma, Berlin (2012)

Saltelli, A., Chan, K., Scott, E.M., et al.: Sensitivity Analysis, vol. 1. Wiley, New York (2000)

Wainer, G.A., Mosterman, P.J.: Discrete-Event Modeling and Simulation: Theory and Applications. CRC Press, Boca Raton (2016)

Ziari, H., Sobhani, J., Ayoubinejad, J., Hartmann, T.: Prediction of IRI in short and long terms for flexible pavements: ANN and GMDH methods. Int. J. Pavement Eng. **17**(9), 776–788 (2016)