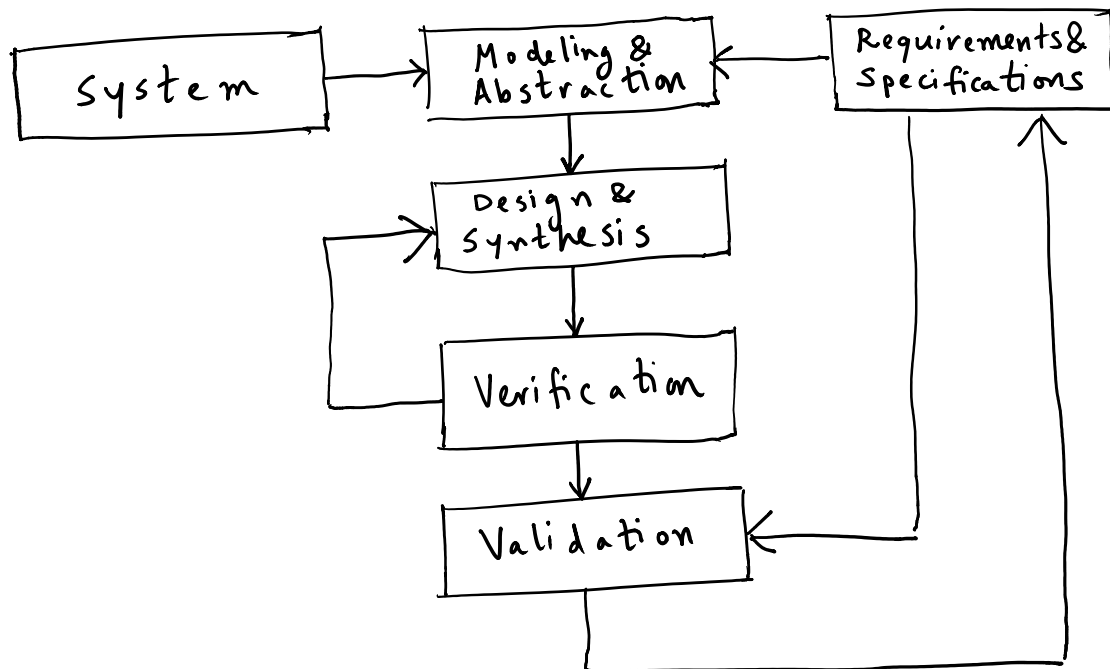

IEOR 151 – Lecture 1

Service Systems

1 Simplified View of Systems Engineering



1.1 MODELING AND ABSTRACTION

An important step in the engineering of service systems is to create an abstraction of both (i) the system and (ii) requirements and specifications, such that the resulting abstraction admits efficient and tractable computation while also remaining expressive enough that the abstraction accurately represents the system response to engineered decisions and its relation to desired behavior. This step is often called modeling, because it can involve matching the system and specifications to common classes of abstract models. In some instances, modeling is also used to refer to using data gathered from the system to identify the parameters (or coefficients) of the corresponding abstract model.

1.2 DESIGN AND SYNTHESIS

Given an abstraction of the system and the desired requirements, another step is the design and synthesis of an intervention for the system to ensure that the requirements and specifications are satisfied. The most significant aspect of this step is computation to generate the intervention, and this leverages methods from mathematical optimization, stochastic processes, and other quantitative approaches. If the model belongs to specific classes, then this step can often be simplified by using appropriate software packages. For this reason, it is common to use choose abstractions partly based on the criteria of the existence of corresponding design and synthesis tools.

1.3 VERIFICATION

Design and synthesis often use abstractions that do not include all aspects of the system behavior, and so it can be the case that a designed intervention will not satisfy the requirements and specifications when applied to the actual system. Given this gap, it is common to perform the step of verification to ensure that the design meets the desired requirements. When deficiencies are found, the step of design and synthesis is repeated to correct any deficiencies. The process of verification is both computational and experimental. For instance, simulation is commonly used to verify systems. Experimental approaches are more costly (in terms of time, cost, and effort) than computational approaches, and so computational verification is typically performed before beginning experimental verification.

1.4 VALIDATION

Even if an engineered system meets the original requirements and specifications, it may be the case that the overall engineered system does not achieve desired outcomes. The step of validation is to understand how well the system meets the needs of the various stakeholders of the engineered system. When the outcomes are identified to be inadequate, one approach to correct the situation is to update the requirements and specifications for the system. Like verification, validation can be both computational and experimental. And though shown as being a subsequent step to verification, it is common to perform computational verification and validation before embarking on experimental verification and validation (V&V).

2 Course Overview

Though this course is nominally about the design and analysis of service operations systems, the steps of (i) abstraction & modeling, and (ii) experimental verification and validation contain subtleties unique service systems. This is not to say that design and synthesis will not be discussed in the course, but rather that design and synthesis will naturally follow once an appropriate abstraction and model is created. With relation to other courses, the computational methods and theory underlying the design and synthesis are based on probability,

mathematical optimization, and other quantitative approaches. Computational verification and validation draws heavily upon simulation, as well as other quantitative approaches.

3 A Restaurant Example

To make the above discussion more concrete, we will describe a coffee shop example that is representative of many of the challenges inherent in design and analysis for service operations systems. Suppose we are analyzing a chain of coffee shops. Customers arrive to the coffee shop at some rate, and then some time is spent in which the customer gives an order and payment to a cashier; additional time is spent by the customer waiting for their drink order to be prepared and served. The question we are interested in answering is: What is the minimum number of cashiers and baristas we should have in each coffee shop to ensure that customers are satisfied?

The first step is how can we abstract this system? One abstraction that will be covered in this course is as a queuing theory model, in which customers stochastically arrive with some average rate, and cashiers and baristas stochastically process orders with some average rate. Within this class of model, one type of specification that can be used is to specify a desired average time spent between (a) arriving in line for the cashier, and (b) receiving a completed order and drink. Given this specification and model (along with quantitative values for the parameters of the abstract model), we can use simple methods to determine the minimum number of cashiers and baristas that are needed. The weakness of this abstraction is that the rate of customer arrivals varies throughout the day, and so we need to validate that our design of the minimum number of cashiers is still sufficient to ensure the desired average time spent by the customers.

Even after we computationally validate our design, it is not clear that the specifications will ensure that customers are satisfied. Measuring satisfaction itself is complex, because there are many different aspects of satisfaction that will be differentially affected by various designs. Imagine that we conduct limited trials in which designs to achieve two different specifications of average time spent by customers are tried on two different sets of representative stores. How can we quantify the differences in customer satisfaction and ensure that these differences are statistically significant, so that we can make a final design choice to ensure customer satisfaction along many different axes and minimize risks due to incorrect design choices?