

Service Engineering: Data-Based Course Development and Teaching*

Avishai Mandelbaum

Faculty of Industrial Engineering & Management,
Technion, Haifa 32000, Israel,
`avim@tx.technion.ac.il`

Sergey Zeltyn

IBM Research Lab, Haifa 31905, Israel,
`sergeyz@il.ibm.com`

August 19, 2009

*There exists a Full Version of the present document [28], which includes expanded descriptions (text, graphs) of lectures and some homework, as well as an additional section on The Fusion of Research and Teaching.

Abstract

In this exposition, we discuss empirically-based teaching in the newly emerging field of *Service Engineering*. Specifically, we survey a “Service Engineering” course, taught at the Technion - Israel Institute of Technology. The course “was born” about fifteen years ago as a graduate seminar in “Service Networks”, continued as an elective course and ultimately took its present form [31], as a core course for the undergraduate program in Industrial Engineering and Management.

The role of measurements and data as teaching-enhancers and research-drivers is underscored. In addition, we emphasize that data granularity must reach the individual-transaction level. We describe customized databases and software tools that facilitate operational and statistical analysis of services; this includes the use of SEESat, a data-user interface that was developed at the Technion’s SEE Laboratory [32], for research and educational purposes. Some unique aspects of the course are the incorporation of state-of-the-art research and real-world data in lectures, recitations and home assignments, as amply presented throughout this work.

The three building blocks for an operational model of a basic service system, as we perceive it, are *arrivals* of service requests, *durations* of service transactions and delay-generated (*im*)*patience* of customers. These are fused, via *service-protocols*, into models of processing/queueing networks which, in turn, form our theoretical framework for Service Engineering. Within this framework, one develops insights, design principles (rules-of-thumb) and tools, for example in support of staffing and controls.

The application focus of the surveyed course has been telephone *call centers*, which constitute an explosively-growing branch of the service industry. Indeed, due to their prevalence and the diversity of their operational problems, call centers give rise to numerous challenges for Service Sciences, Engineering and Management. The course is now expanding to also cover healthcare, especially hospitals; some examples from other service areas (e.g. the justice system) are described as well.

Contents

1	Introduction to Service Engineering	4
1.1	Motivation and Contents of the Paper	4
1.2	Course Homepage(s)	5
1.3	Service Networks: Models of Congestion-Prone Service Operations	6
1.3.1	On Queues in Service	6
1.3.2	On Service Networks and their Analysis	6
1.4	Some Relevant History of Queueing-Theory	8
1.4.1	The Early Days	8
1.4.2	QED Queues	9
1.4.3	ED Queues	10
1.4.4	Summary	10
1.5	Service Engineering: Challenges, Goals and Methods	11
1.5.1	Challenges and Goals	11
1.5.2	Scientific Perspective	12
1.5.3	Engineering Perspective	12
1.5.4	Phenomenology, or Why Approximate	13
2	Course Goals and History	14
3	Data - a Prerequisite for Research and Teaching	14
3.1	DataMOCCA - Model for Measurements from Service Systems	16
4	Course Syllabus: Theory, Examples, Case Studies	17
4.1	Course Material and Supporting Texts	18
5	ServEng Homework and Exams: A Data-Based Approach	24
5.1	Homework Assignments	24
5.1.1	List of Assignments	25
5.2	The Final Exam	26
6	Acknowledgements, and a Little More History	27

1 Introduction to Service Engineering

1.1 Motivation and Contents of the Paper

The service sector is central in the life of post-industrial societies - more than 70% of the Gross National Product of most developed countries is produced by this sector. (See Fitzsimmons and Fitzsimmons [9] for background on Services.) In concert with this state of affairs, there exists a growing demand for high-quality multi-disciplinary research in the field of services, as well as for a significant number of *Service Engineers*, namely scientifically-educated specialists that are capable of designing service systems, as well as solving multi-faceted problems that arise in their practice.

In the U.S., the academic home for the area of Services has traditionally been the Business School, where Services have been taught most often as Service Marketing. Another business school option is Service Management, either within Operations Management courses or, rather rarely, as a stand-alone Service Operations course. As an engineering discipline, the natural home for Services are Industrial Engineering units. Indeed, our original use of the term “Service Engineering” was conceived by combining the relevant words in “Service Management” and “Industrial Engineering”. We had roughly in mind a “*New Age Industrial Engineer* that must combine technological (and scientific) knowledge with process design in order to create (and operate) the (service) delivery systems of the future [12].”

It is our belief that there exists a broad gap between academia’s supply and the demand for Service Science and Engineering. Focusing on the education, universities either do not offer Service Engineering courses or, when they do so, the education quality is typically not at par with that of the traditional engineering disciplines, in particular that provided by leading Industrial Engineering units.

The goal of the “*Service Engineering*” (ServEng) course, “born” at the Technion - Israel Institute of Technology about fifteen years ago [31], is to narrow down the educational gap between demand and supply, while also stimulating Service research. As we perceive it, the ultimate goal of Service Engineering is to *develop scientifically-based design principles and tools (often culminating in software), that support and balance service quality, efficiency and profitability, from the likely conflicting perspectives of customers, servers, managers, and often also society*. The goal of our ServEng course is more restricted: we take an Operations point of view, hence we focus on *operational service quality*, on *service efficiency* and the tradeoffs between the two.

Our ServEng course takes a *data-based approach to teaching*. Thus, throughout the paper, course contents will be illustrated with the help of practical examples, drawing from data-based case studies. Students encounter such case studies continuously during lectures,

recitations and homework assignments. For example, a concept can be illustrated in a recitation via data from bank-tellers' service, homework practices the concept on call center data, and students are tested on it using data from an emergency department.

The remaining part of Section 1 is dedicated to our paradigm of the Service Engineering discipline. Specifically, Section 1.2 presents the course homepage, where the ServEng materials can be downloaded. Section 1.3 introduces Service Networks as our modeling framework for a service operation, with queueing theory and science constituting the main theoretical foundation. Section 1.4 elaborates on relevant milestones in the evolution of Queueing Theory. Then, Section 1.5 focuses on the main challenges, goals and methods in the ServEng discipline, in the context of the course. In this part, we also explain the need for approximations (compromises) in modeling and analysis of service systems.

Section 2 elaborates on the goals of the ServEng course and provides a brief survey of its history. Section 3 explains our data-based teaching approach and introduces the DataMOCCA (Data Model for Call Center Analysis) system: through its graphical user interface, SEESat, this system enables a friendly effective access to several large data repositories from service systems (currently call centers and emergency departments), which have been used for research and teaching (specifically in lectures, recitations and homework of the ServEng course).

In Section 4 we briefly describe the lectures of the course. Section 5 lists the data-based home assignments, where students must use DataMOCCA/SEESat tools for the analysis of call center data¹. We conclude, in Section 6, with acknowledgements to those who helped bring the course to where it is now, interwinding it with some additional course history.

1.2 Course Homepage(s)

Before continuing, readers are advised to browse through the ServEng website, and get an idea of its structure and contents. References to relevant links will be provided as we go along. For example, the "Table of Contents" of the recently taught course is in <http://iew3.technion.ac.il/serveng2009S/Lectures/Schedule1.doc>.

The formal course website is ie.technion.ac.il/ServEng, which includes the material of a *complete* course - the last one to have been taught. Then, as the course is being offered, a second website is constructed gradually and updated as the course progresses; the address of this dynamic website is appended by the course's year and semester. For example, the site of the course taught during the Spring semester of 2009 has the address ie.technion.ac.il/ServEng2009S. (The earlier Winter semester had 2009W appending its address).

¹An expanded description of the lectures, as well as some homework, appears in our Full Version [28].

1.3 Service Networks: Models of Congestion-Prone Service Operations

The title of this section reflects our angle on service operations - we often view them as stochastic (random) or deterministic (fluid) systems, within the Operations Research paradigm of Queueing Networks. To support this view, let us first present our conception of the roles of Queues in services, from the perspectives of customers, servers and managers. We shall then describe Service Networks, continuing with some relevant queueing-theory history.

1.3.1 On Queues in Service

Queues in services are often the arena where customers, service-providers (servers) and managers interact (establish contact), in order to jointly create the service experience. Process-wise, queues play in services much the same role as inventories in manufacturing (see JIT = Just-in-Time, TBC = Time-based-Competition, etc.). But, in addition, “human queues” express preferences, complain, abandon and even spread around negative impressions. Thus:

- *Customers* treat the queueing-experience as a window to the service-providing party, through which their judgement of it is shaped for better or worse.
- *Servers* can use the queue as a clearly visible proxy for the state of the system, based on which, among other things, service protocols can be exercised (eg. customers priorities).
- *Managers* can use queues as indicators (queues are the means, not the goals) for control and improvement opportunities. Indeed, queues provide unbiased quantifiable measures (these are not abundant in services), in terms of which performance is relatively easy to monitor and goals (mainly tactical and operational, but sometimes also strategic) are naturally formulated.

Our point of view is thus clear: the design, analysis and management of queues in service operations could and should constitute a central driver and enabler in the continuous pursuit of service quality, efficiency and profitability.

1.3.2 On Service Networks and their Analysis

Service Networks here refer to *dynamic (process)* models (mostly analytical, sometimes empirical, and rarely simulation) of a service operation as a *queueing network*. The dynamics is that of serving *human customers*, either directly face-to-face or through phone-calls, email,

internet etc. Informally, a *queueing network* can be thought of as consisting of interconnected service stations. Each station is occupied by servers who are dedicated to serve customers queued at the station.

In its simplest version, the evolution of a service station over time is stationary, as statistically-identical customers arrive to the station either exogenously or from other stations. Upon arrival, a customer is matched with an idle server, if there is any; otherwise, the customer joins a queue and gets served first-come-first-served. Upon service completion, customers either leave the network or move on to another station in anticipation of additional service. Extensions to this simplest version cover, for example, models with non-stationary arrivals (peak-loads), multi-type customers that adhere to alternative service and routing protocols, customer abandonment while waiting (after losing their patience), finite waiting capacities that give rise to blocking, splitting (fork) and matching (join) of customers, and much more.

In analyzing a Service Network, we find it useful to be guided by the following four steps (though, unfortunately, most often only the first three are applied or even applicable):

- *Can we do it?* Deterministic capacity analysis, via process-flow diagrams (spreadsheets, linear programming), which identifies resource-bottlenecks (or at least candidates for such) and yields utilization profiles.
- *How long will it take?* Typically stochastic response-time analysis, via analytical queueing network models (exact or approximate) or simulations, which yields congestion curves. Note that when predictable variability prevails and dominates, then the “fluid view” and the corresponding deterministic methods are appropriate; see Section 4.4 of Full Version [28].
- *Can we do better?* Sensitivity and Parametric (“what-if”) analysis, of Measures of Performance (MOP’s) or Scenarios, which yields directions and magnitudes for improvements.
- *How much better can we do?* or put simply: “What is optimal to do?” This type of questions has been typically addressed via Optimal Control (exact or asymptotic) which, as a rule, is difficult but becoming more and more feasible. (Another developing research direction is optimization via simulation - see [7] for a call-centers example, which is now in the process of being incorporated into ServEng.)

Several examples of service networks are covered in Full Version [28]; see, for example, the Dynamic-Stochastic (DS) PERT/CMP networks (sometimes referred to as fork-join or split-match networks) in Section 4.3.

1.4 Some Relevant History of Queueing-Theory

Queueing theory provides a central mathematical foundation for ServEng. It is thus appropriate to cover some relevant milestones in this theory's evolution.

1.4.1 The Early Days

The father of Queueing Theory was the Danish telecommunication engineer Agner Krarup *Erlang* who, around 1910-20, introduced and analyzed the first mathematical queueing models. Erlang's models [5] are standardly taught in elementary/introductory academic courses (for example $M/M/n$, $M/M/n/n$), as they are still corner-stones of today's telecommunication models (where $M/M/n/n$ is known as Erlang-B, "B" apparently for Blocking - the central feature of this model, and $M/M/n$ is referred to as Erlang-C, "C" conceivably because it is subsequent to "B"). Moreover, and more relevant to our present discussion, $M/M/n$ is still the work-horse that supports workforce decisions in many telephone call centers; see Section 4.6 of Full Version [28].

Another seminal contributor to Queueing Theory, Scandinavian (Swedish) as well, is Conny Palm, who in 1940-50 added to Erlang's $M/M/n$ queue the option of customers abandonment [30]. We shall refer to Palm's model as Palm/Erlang-A, or just Erlang-A for short (unfortunately and rather unjustly to Palm, but Erlang "was there first".) The "A" stands for Abandonment, and for the fact that Erlang-A is a mathematical interpolation between Erlang-B and Erlang-C. Palm, however, has been mostly known for his analysis of time-varying systems, also of great relevance to service operations and hence covered, in some way, in ServEng.

A next seminal step (one might say a "discontinuity" in the evolution of Queueing Research) is due to James R. Jackson, who was responsible for the mathematical extension of Erlang's model of a *single* queueing-station to a system of *networked* queueing stations, or Queueing Networks, around 1955-1965. Jackson was motivated by manufacturing systems and actually analyzed open and semi-open networks. Closed networks, relevant to healthcare and call centers with an answering machine, as it turns out, were analyzed in the mid 60's by William J. Gordon and Gordon F. Newell. Interestingly, Newell was a Transportation Engineer at Berkeley and also the earliest influential advocator of incorporating Fluid Models as a standard part of Queueing Theory - see his text book [29]. A student of Newell, Randolph W. Hall, who is currently a Professor at University of Southern California, wrote an excellent Queueing book [16] that has influenced our teaching of Service Engineering; Hall is currently working on healthcare systems, adopting the fluid-view (described below) to model the flows of patients in hospitals; see [17].

Jackson networks are the simplest theoretically tractable models of queueing networks.

(Their simplicity stems from the fact that, in steady state and at a fixed time, each station in the network behaves like a naturally-corresponding birth-death model, *independently* of the other stations.) The next step beyond Jackson networks are BCMP/Whittle/Kelly networks, where the heterogeneity of customers is acknowledged by segregating them into classes. But service operations often exhibit features that are not captured by Jackson and BCMP/Whittle/Kelly networks. Further generalizations are therefore needed, which include precedence constraints (fork-join, or split-match networks), models with one-to-many correspondence between customer types and resources (skills-based routing, agile workforce), and models that exhibit transient behavior (as opposed to steady-state analysis).

1.4.2 QED Queues

The key tradeoff in running a service operations is that between service efficiency and quality, which queueing models are ideal to capture. This tradeoff is most delicate in large systems (many servers), but here exact analysis of queueing models turns out limited in its insight. This was already recognized by Erlang around 1910, and later by Pollaczek around 1930, who both resorted to approximations. However, the first to put operational insights of many-server queues (as relevant to us) on a sound mathematical footing were Shlomo Halfin and Ward Whitt, at the early 80's, in the context of Erlang-C (and its extension GI/M/n). They introduced what we shall call QED Queues [15], which stands for queues that are *both* Quality- and Efficiency-Driven, hence their name.

QED Queues emerge within an asymptotic framework that theoretically and insightfully supports the analysis of the efficiency-quality tradeoff of *many-server* queueing systems. The theory culminates in a simple rule-of-thumb for staffing, the square-root staffing rule: if the offered load is R Erlangs (R =arrival rate times average service time), then

$$n \approx R + \beta\sqrt{R}$$

is a staffing level that would appropriately balance quality and efficiency. Here β is a constant, which corresponds to grade-of-service; its specific value (which is practically small - less than 1.5 in absolute value) can be determined by economic considerations (for example the ratio between delay costs and staffing costs).

Prime examples of QED queues are well-run telephone call centers; but to properly model these, one must generalize the Halfin-Whitt framework to allow for customers' impatience. This was done in a Technion M.Sc. thesis by Ofer Garnett, in the late 90's, and later published in [14]. At that same time, a generalization of the Halfin-Whitt framework to time-varying Jackson-like networks was carried out with Bill Massey and Marty Reiman [24] (under the name Markovian Service Networks). In analogy to QED queues, there are also

ED (Efficiency-Driven) and QD (Quality-Driven) queues, all arising from asymptotic analysis as well: Erlang-C, in these three operational regimes, was treated in [1]; generalizations to Erlang-A and relatives is the subject of the Ph.D. thesis of the second author (S.Z), published in [38]. The research-part of the website of Ward Whitt is recommended for further references on QED/ED/QD queues [36].

QED approximations turn out to be amazingly accurate and robust. This arose explicitly in [1], where *square-root staffing* was shown to be an asymptotically optimal staffing level (n in an $M/M/n$ queue). It turns out that the square-root recipe rarely deviates from the actual optimal value by more than 1 server, over a wide range of staffing levels - from the very few (10 or so) to the very many (1000's). This unexpected accuracy has been recurring for many other models since, and it can now be explained theoretically for the Erlang C/B/A models, as well as $M/D/n$; see, for example, [19]. The accuracy of QED approximations has practical significance since it expands the scope of these approximations to relatively small systems - in particular healthcare systems (e.g. emergency departments and internal wards). This was first observed by Otis Jennings and Francis de Vericourt [20]. It is further expanded on in the ongoing PhD thesis of Galit Yom-Tov [37], which supports decisions on static capacity (hospital beds) and dynamic capacity (e.g. nurses). Interestingly, Galit's basic queueing model for beds+nurses is the one that also appeared in the M.Sc. thesis of Polyna Khudyakov [21] on call centers with an answering machine - indeed, beds correspond to trunk-lines and nurses to telephone tellers.

1.4.3 ED Queues

A balance between service quality and efficiency is often desirable, yet sometimes it is infeasible, or perhaps even not optimal. Then one settles for an ED (Efficiency-Driven) service system, in which the focus is on high utilization of resources (the servers), at the cost of service quality (relatively long delays). In call centers, ED performance arises when resources are restricted (e.g. government services); however, with large enough of a system, service level could still be acceptable or even better. (See Ward Whitt's homepage [36] for theoretical papers on many-server ED Queues.) But ED performance is in fact prevalent in healthcare, which renders small-server heavy-traffic asymptotics more relevant - see, for example, the ongoing M.Sc. thesis of Asaf Zviran [39], on fork-join queues in heavy-traffic.

1.4.4 Summary

To summarize, queueing networks have been successfully used to model systems of manufacturing, transportation, computers and telecommunication. For us they serve as indispensable models of service systems, in which customers are human and queues, broadly

interpreted, capture prevalent delays in the service process. The service interface could be phone-to-phone (naturally measured in units of seconds), or face-to-face (in minutes), fax-to-fax (hours) letter-to-letter (days), face-to-machine (e.g., ATM, perhaps also Internet), etc. The finer the time-scale, the greater is the challenge of design and management. Accordingly, the greater is the need for supporting rigorous models, a need that further increases with scale, scope and complexity.

1.5 Service Engineering: Challenges, Goals and Methods

We have been advocating the terminology “*Service Engineering*” to describe our research, teaching and consulting on (tele-)services. (Service Engineering is to be compared against the traditional *Industrial Engineering*, and it is to provide an essential support and supplement to *Service Management*.)

1.5.1 Challenges and Goals

Research, teaching and practice of Service Engineering, as we perceive it, should take a *designer’s view*. Design challenges pertain, for example, to the following issues.

- Service strategy: determinants of service-quality levels, full- versus self-service, customization versus standardization, warranty (after-sales support depth), etc.
- Choosing the Service Interface (channel): assume that service can be potentially performed by phone and/or by email, fax, letter, or perhaps face-to-face.
- Designing Service Process: should one use front- or back-office (or possibly both), should the tasks be performed sequentially or in parallel, etc.
- Control issues: which customers to admit to service, priority scheduling, skills-based-routing, exploiting idleness, etc.
- Operational resource management: how many servers to staff in order to reach an acceptable service level (with staffing being performed off-line or on-line, the latter as a corrective action to inaccurate forecasting), shifts scheduling, etc.
- Designing the service environment: waiting experience, using busy-signal versus music in call centers, providing information to customers (eg. predicting delay durations), etc.
- Marketing: customer segmentation, cross- or up-selling, marketing-operations interfaces, etc.

- Information Systems: data-base design of call-by-call operational and business data, off-line and on-line queries, etc.
- Taking into account human factors: career paths, incentives, hiring policies, number of call center agents who are physically present at the workplace versus actual workforce level (number of agents who are actually available to take calls).

The above designer’s view supports our vision of the ultimate goal of Service Engineering (both in research and teaching), as stated in Section 1.1: to develop scientifically-based design principles and tools that support and balance service quality, efficiency and profitability. We find that queueing-network models constitute a natural convenient nurturing ground for the development of such principles and tools. However, the existing supporting (Queueing) theory has been somewhat lacking, as will now be explained.

1.5.2 Scientific Perspective

The bulk of what is called Queueing Theory (recall Section 1.4) consists of research papers that formulate and analyze queueing models with realistic flavor. Most papers are knowledge-driven, where “solutions in search of a problem” are developed. Other papers are problem-driven, but most do not go far enough to a practical solution. Only relatively few articles develop theory that is either rooted in or actually settles a real-world problem, and scarcely few carry the work as far as validating the model or the solution. In concert with this state of affairs, not much is available of what could be called “*Queueing Science*”, or perhaps the Science of Congestion, which should supplement traditional Queueing Theory with data-based models, observations and experiments. In service networks, such “Science” is lagging behind that in telecommunications, transportation, computers and manufacturing. Key reasons seem to be the difficulty to measure services (any scientific endeavor ought to start with measurements), combined with the need to incorporate human factors (which are notoriously difficult to quantify). Since reliable measurements ought to constitute a prerequisite for proper management (see TQM = Total-Quality-Management, for example), the subject of measurements and proper statistical inference is important in our context.

1.5.3 Engineering Perspective

Service networks provide a platform for advancing, what could be described as, Queueing Science and Management Engineering of Sociotechnical Systems. Management Engineering links Management Science with Management Practice, by “solving problems with existing tools in novel ways” [3]. Quoting the late Robert Herman [18], acknowledged as the “father

of Transportation Science”, Sociotechnical systems are to be distinguished from, say, “physical and engineering systems, as they can exhibit incredible model complexity due to human beings expressing their microgoals”. (Significantly, Herman’s models of complexity were nevertheless “tractable through remarkable collective effects”; in other words “laws of large numbers” which, for services as well, turn out to play a central explanatory role.) The approach and terminology that we have been using, namely *Service Engineering*, is highly consistent with the once influential BPR (=Business-Process-Reengineering) evolution, as well as with ERP (=Enterprise-Resource-Planning) and CRM (= Customer-Relations-Management, or the more rational acronym Customer-Revenue-Management), placing heavy emphasis on the process-view and relying heavily on the accessibility of information technology.

1.5.4 Phenomenology, or Why Approximate

Service systems often operate over finite-time horizons (the notion of steady-state then requires re-interpretation). They employ heterogeneous servers, whose service capacities are time and state-dependent. Their customers are “intelligent”, who typically (but not always) prefer short queues; they jockey, renege and, in general, react to state-changes and learn with experience. Finally, service systems suffer from high variability – both predictable and unpredictable, and diseconomies of scale – when being decentralized and inefficient (e.g., often FCFS/FIFO is the only option). Such features render the modeling of service networks a challenge and their exact analysis a rarity. This leads to research on *approximations*, typically short but also long-run fluid and diffusion approximations. Approximations also enhance exact analysis by simplifying calculations and exposing operational regimes that arise asymptotically.

The “ultimate products” of approximations are scientifically-based practically-useful rules-of-thumb. Examples of such rules-of-thumb will be demonstrated below. For example, in Section 4.7 of Full Version [28] on staffing, we cover staffing rules, customized to the QD/ED/QED operational regimes; in particular, square-root staffing corresponds to the QED regime. (Recall the discussion on these regimes in Section 1.4.)

Our approximations also reveal the “right” scale (time and space) at which phenomena evolve. For example, in ED many-server systems, waiting time is expected to be of the order of a service time. And in QED many-server systems, waiting time is expected to be one order of magnitude less than service time. To be concrete, in QED call centers, waiting is measured in seconds vs. service times in minutes; in transportation, time to seek parking in a busy city down-town is measured in minutes while parking time in hours; and in emergency departments, waiting to be hospitalized is measured in hours vs. hospitalization time in days. As another scaling example, QED call centers enjoy abandonment rates that

are inversely proportional to the square-root of the number of agents; this enables one to draw performance curves, as in Figure 27 in Full Version, where a *single* graph predicts abandonment rates for call centers of *all* sizes.

2 Course Goals and History

The Service Engineering (ServEng) course has been taught for nearly fifteen years at the Faculty of Industrial Engineering and Management (IE&M) in the Technion - Israel Institute of Technology. It started as a seminar for graduate students, entitled “Service Networks”. It had then evolved through the stages of a graduate course that can be attended by advanced undergraduate students, to an elective undergraduate course, attended also by graduate students who seek an introduction to the field of Service Engineering. Currently, Service Engineering is an undergraduate *core* course, taught each semester and attended by approximately 100 students yearly. (Many of the IE&M graduate students in Stochastic Operations Research and Statistics take it as well.) The ServEng website [31] is designed to contain all course materials (lecture notes/slides, recitations, homework), supplemented by related research papers, slides of seminars, software, databases and more.

As discussed in Section 1.1, the service sector is dominating the economics of developed countries. Still, prior to the launch of the ServEng course, Technion IE&M students had been exposed mainly to methods and techniques inspired by manufacturing applications. (This situation seems to prevail also among IE departments at other universities.) The ServEng course aims at filling this gap by providing students with appropriate models and tools for design, operation and analysis of service systems. Our teaching approach is data-oriented: examples from various service sectors are presented at lectures, recitations and home assignments, with the call center industry being the central (but in no way unique) application area.

In one and a half decade of course development, our service system data repositories have been developed in parallel with the theoretical material of the course. In the next section, we describe the service system data that supports the course.

3 Data - a Prerequisite for Research and Teaching

We strongly believe that systematic measurements and data collection are prerequisites for the analysis and management of any service system. Therefore, our students encounter numerous service data sets and examples during lectures and recitations. Most of the course homework is also based on actual service data.

Early generations of the course used one-month tellers' data from a bank in Israel, in support of recitations and homework. As described in [26], the data was collected through bar-codes that were given to customers upon arrivals, which were then scanned at milestones of the service experience. Then the focus of the course shifted in the direction of call centers, which seemed to be a natural candidate for becoming our main application area. First, one could not but be impressed by the sheer magnitude of the call center industry, with its explosive growth over the last three decades. Millions of agents are employed in call centers over the world and, according to some estimates, worldwide expenditure on call centers exceeds \$300 billion. Second, the call center environment gives rise to numerous managerial and engineering challenges that vary in their nature and time-scale, from real-time skill-based routing calls to long-term design and workforce planning. And thirdly, large call centers generate vast amounts of data. A detailed history of each call that enters the system can, in theory, be reconstructed via the Automatic Call Distributor (ACD) and Interactive Voice Response Units (IVR). However, call centers have not typically stored or analyzed this data, using instead the ACD reports that summarize performance over certain time intervals (say, 30 minutes). In our research [2] and teaching, we advocate the change of this approach and emphasize the practical and research advantages of call-by-call data analysis.

In the ServEng course, our first call-center data-base covered a one-year operation of a call center at a small Israeli bank [2, 25]: about 350,000 calls, catered by about 15 agents. Once this database was incorporated into the course, the bank tellers' data (face-to-face services) has been since used exclusively in recitations, to teach or demonstrate techniques, while the telephone data has been used in homework, in order to practice these techniques.

Our small Israeli database clearly revealed the potential of incorporating real-world data into ServEng teaching and research. It hence paved the way to access vast amounts of call-by-call data, from large call centers who were willing, sometimes even eager, to participate in our data venture. But then several challenges arose. First, call center data is processed by vendor-specific programs, in formats (data-structures) that are typically not amenable to operational analysis. Second, a database of a large call center consists of up to hundreds of millions of records. Hence, brute force statistical processing of the records would take a prohibitively long time even for the most powerful statistical software. And thirdly, our experience is that a significant part of real-world databases is contaminated or inconsistent with the rest, which calls for an extensive cleaning and data-improvement effort prior to using the data.

As one anecdotal example, the first large US Bank that provided us with call center data had four interconnected call centers. When switching to daylight saving time, one of these call centers failed to update its computer clock and thus, calls that were dialed in City X

arrived to City Y one hour earlier! The database from this U.S. Bank, which is described in [35], was in fact cleaned and has recently been incorporated into the ServEng course, as will be described in the next subsection. (Just for comparison sake, U.S. Bank employed around 1000 call center agents, who catered to about 350,000 calls per *week* - which was the yearly volume of our small Israeli bank.)

3.1 DataMOCCA - Model for Measurements from Service Systems

To address the challenges that databases of large call centers present, a research laboratory was established at the Technion: SEELab, where SEE stands for “Service Enterprise Engineering” [32]. The SEELab, under the supervision of Dr. Valery Trofimov, *receives, processes, cleans, validates and maintains* a repository of databases from service enterprises. To this end, the SEELab developed DataMOCCA (Data Models for Call Centers Analysis) [33], which is a software suite that renders the data ready for use in research and teaching.

DataMOCCA offers a universal model for operational call center data. Its main interface with its users is a (powerful and friendly) graphical user-interface, SEEStat, that enables real-time statistical analysis, at second-to-month resolutions. Beyond empirical analysis, SEEStat offers statistical algorithms such as parametric distribution fitting, fitting of distribution mixtures, survival analysis and more - all these algorithms interact smoothly with all the databases.

SEE’s databases are designed and maintained at two levels: the basic level is close to the raw data, after cleaning, validating and processing; the upper level consists of precompiled summary tables, which are created only once, and they are efficient enough to support real-time processing, with few-seconds response time, covering gigabytes-large databases. This creates a convenient environment that accommodates real-time statistical analysis and simulations.

Currently, DataMOCCA covers call-by-call data of three large call centers: a U.S. bank, an Israeli bank and an Israeli cellular-phone company, over periods of 2-3 years each. (For example, the U.S. bank data has close to 220 million calls, out of which about 40 million were served by agents and the rest by a VRU - Voice Response Unit.) DataMOCCA and SEEStat have well-served our ServEng course. Indeed, a multitude of example, using SEEStat, are presented in classes and recitations. Moreover, starting last year, one of the homework assignments is SEEStat-based, providing the students with a working copy of the software. This copy is connected to 1.5 years worth of data from the U.S. Bank mentioned above (its database is available for public use), and it enables the students to analyze the data in interesting insightful ways. Concrete applications of SEEStat, from the ServEng course, will

be demonstrated below, mostly in Section 4.

While originally designed for operational databases from call centers (hence its name), DataMOCCA has been generalized to accommodate additional sources and types of data. Specifically, we now have data also from several hospitals (mostly their emergency departments) and from an internet website (click-stream data). We are further preparing to augment our operational databases with financial and contents data. (In an emergency department, contents refers to clinical data). In addition, two simulators have been written, one for a call center and the other for an emergency department. The “vision” is to connect these simulators to the real data at the SEELab, and then have all SEELab resources (data and simulators) readily accessible worldwide, for use by students and researchers of Service Engineering and its supporting Service Science.

4 Course Syllabus: Theory, Examples, Case Studies

The ServEng course consists of roughly four parts:

1. *Prerequisites*: measurements and models.
2. *Building Blocks*: demand, services, customers (im)patience;
3. *Models*: deterministic (Fluid) and stochastic - mainly queueing models, both conventional (Markovian) and approximations;
4. *Applications*: design, workforce-management (e.g. staffing) and skills-based routing.

We now provide a brief description of these four parts. We continue with a list of the lectures, each accompanied by brief commentary. In our Full Version [28], readers can find more detailed descriptions of each of the 14 ServEng lectures.

Measurements, at the granularity level of individual service transactions, are prerequisites for the design, analysis and management of service systems. Thus, after opening the course with an introduction to Service Engineering, we survey transactional measurement systems in face-to-face, telephone, internet and transportation systems. We then proceed with an introduction to *Modeling*, using Dynamic Stochastic PERT/CPM models (also called fork-join or split-match networks) as our modeling framework. These models capture operational congestion that is due to resource constraints and synchronization gaps.

Measurements and modeling prerequisites directly give rise to deterministic (fluid/flow) models of a service station, which capture average behavior and enable relatively simply yet far-reaching analysis - for example capacity (bottleneck) analysis.

The next course segment is dedicated to the three building blocks of a basic service-model: demand, service and (im)patience. First we study *service demand*, emphasizing the importance of reliable forecasting techniques. (In particular, our model for exogenous customers' arrivals is a Poisson process, or a relative). Then we analyze the *service process*, describing its operational characteristics: service-duration, which is a static characteristics, and process structure, capturing dynamic evolution. Service durations in call centers often turn out log-normally distributed [2] (at the resolution of seconds, but sometimes exponential at minutes-resolution). The structure of the service process is naturally captured by phase-type distributions. We end with *customers' patience*, or perhaps *impatience*, and its manifestation - the abandonment phenomena, which is important in call centers and other services (e.g. Internet and even Emergency Rooms).

The three building blocks, of arrivals, services and (im)patience, are fused into basic queueing models where customers are i.i.d. and servers are also i.i.d. A central role is played by Markovian Queues, underscoring the applicability of the Erlang-A queue in call center industry [27]. Then we discuss design principles (pooling to exploit economies of scale) and present operational workforce management techniques (staffing and scheduling), including staffing in the QED, ED and QD operational regimes. We conclude the course with models that acknowledge customers differentiation (priorities) and servers heterogeneity/skills (SBR = Skills-Based-Routing). An optional last lecture surveys queueing networks, specifically Jackson and Generalized Jackson, as models of multi-stage service systems.

4.1 Course Material and Supporting Texts

ServEng is a one-semester course. Our standard Israeli semester consists of 14 weeks, three-hour lectures per week, accompanied by a weekly one hour recitation. At every lecture and recitation, students get lecture notes (copies of the slides) which are used in class. Typically, these notes are supplements with board lectures, providing an introduction to a sub-subject or clarifying subtle issues. (The notes most often constitute a superset of the class material, and the actual material covered varies.) All lecture notes appear in the ServEng website [31], under the Lectures and Recitations menus.

Israeli students, unlike U.S. students, are not accustomed to using text-books for self-study - they rely on class material, hence the lecture notes play a central role in the course. Yet, there are three books that are mentioned as relevant and useful, and our library has ample copies of these books for students' use. These books are mentioned at the outset of the course, and students are asked to read few chapters from these books. The books are by Randolph Hall [16], J. Fitzsimmons and M. Fitzsimmons [9] and C. Lovelock [23] (in that order of relevance to the course), which we now elaborate on.

Our teaching philosophy was influenced by Hall's book [16] on Queueing Methods - this book is rather unique among Queueing books as it discusses measurements, gives fluid models the respect they deserve and emphasizes science-based applications. Hall's book serves as an optional textbook, which students are encouraged to consult, especially during the first half of the course. Two additional background-textbooks are by Fitzsimmons and Fitzsimmons [9] and Lovelock [23]. Placing these books in perspective, Hall's book lies at the intersection of Operations Research and Industrial Engineering; Fitzsimmons and Fitzsimmons borders on Industrial Engineering and Operations Management; and Lovelock touches on the four extreme points that span Service Engineering: Operations, Marketing, Human Resource Management and Information Systems.

We now proceed with listing the topics that we cover, plus some commentary.

Rules of the Game: In the first lecture of the course, we introduce students to the Service Economy and to the discipline of Service Engineering. We also discuss course logistics and "rules of the game", for example the relative weights of homework (40-50%) and final exam (60-50%) in the final grade.

Measurements and Models; Little's Law: The subsequent two lectures are dedicated to *measurements* and *models* - these, in our opinion, are the prerequisites for any advances and practice of Engineering and Science, in particular to the discipline that we are teaching.

Data Sources: As already mentioned, we believe that research and teaching of Service Engineering must be based on *measurements*: real-world data, collected in various service systems at various levels of granularity. All stages of modeling and design of service systems must be empirically supported, starting with systematic measurements and data collection at the initial stage of modeling and ending with the validation of both models and managerial decisions against empirical data. Call Centers provide an excellent illustration for our approach, and their operational data, at the level of individual transaction (*transactional data*), has been our main data source. Recently, we have extended our data sources to cover also hospitals, especially their emergency departments. Additional course data comes from face-to-face, internet, and transportation services.

Models: The second prerequisite for Service Engineering is *models*. Indeed, a major part of the course is dedicated to different models of service systems, their analysis and areas of application. We distinguish between *empirical* and *analytical* models. (Presently, the course utilizes no *simulation* models, though this is going to change in the near future.) Empirical models, which are directly data-based, are further classified into conceptual, descriptive and explanatory models. Analytical (or mathematical) models are either deterministic (fluid-based) or stochastic (e.g. Markov chains). All models could be either static (long-run,

steady-state) or dynamic (transient). No attempt is made at the course to well-define the above mentioned model classes - these are used merely to structure one’s thinking about modeling, and we “explain” them mainly through case studies. For example, some fluid models are in fact empirical models, as we clarify through Figure 11 in Full Version [28]. And analytical models could overlap with empirical models, as we illustrate via the simple yet fundamental Little’s Law [22]: $L = \lambda \times W$. (The reality of service systems, for example transient behavior, calls for versions and generalizations of Little’s Law that go beyond steady-state/long-run. We thus pay a significant attention to *time-varying* versions, which includes *finite horizons* as well.)

Addressing the Skeptic: Is Service Engineering Relevant? It is not unreasonable to question whether the methods of “exact sciences” (eg. Mathematics), and their related models (Statistical, Operations Research), are relevant for supporting the operations of the legal system. Indeed, Flanders [10], who is what we have been calling a *smart influential skeptic*, gives a negative answer to this question. Interestingly, Flanders demonstrates unfamiliarity with Little’s Law (bottom of page 316 and top of 317; Note 5. at paper’s end). We thus believe that the following case study of Little’s Law, applied to the “Production of Justice”, would have certainly helped a Service Engineer alter the views of any “Flanders”, in favor of the profession.

Case Study - Little’s Law and the “Production” of Justice: This example, presented in Section 4.2.4 of Full Version [28], applies Little’s law to data that records the operational performance of five judges in Haifa’s labor court. It demonstrates that the *best* operator (judge), having the highest λ and lowest W (i.e. processing most files with the least delay), has in fact the largest L (longest queue). Hence, according to the accepted criterion of the justice system, it is considered the *worst* operational performer, having the longest files in process.

Little’s Law for Time-Varying Systems: Towards the end of the course, we teach staffing algorithms for service systems that face time-varying arrival rates (which excludes little else). These algorithms are based on the central notion of *offered load* which, in turn, amounts to a version of Little’s Law in a *time-varying* environment. We introduce this time-varying version, followed by discussions of the offered load and the relation of the two.

The Offered-Load: For a service system in steady-state, with constant rate λ , and iid services of (random) durations S , the *arrival-rate of work* to the system is $R = \lambda \cdot E[S]$ units of work per unite of time, where work is measured in units of time. We call R the *offered-load* to the system. What if the arrival rate λ varies in time? The “right” answer conceptualizes the workload offered to a service station in terms of a corresponding infinite-server system,

which is natural and far-reaching. The offered-load is the backbone of time-varying staffing (taught at the end of the course). The concept, which is presently the subject of active research at the Technion, extends far beyond a single basic service station.

Dynamic Stochastic Service Networks: In Lecture 4, we formally introduce Service Networks (or processing networks), which were described previously in Section 1.3. These provide us with a framework for teaching the main drivers of operational delays: *scarce resources* and *synchronization gaps* (to be distinguished from other causes, beyond the control of the Service Engineer, for example bad weather in an airport). We actually start with loosely-defined conceptual models of service networks, made concrete through Call Centers and Emergency Departments. These conceptual models demonstrate (at least) two important points: the power of the “language of modeling”, and the role of the Service Engineer as an *integrator* of several disciplines (recall [12]).

Applicable Conceptual Framework: DS-PERT networks serve as a conceptual framework that teaches students at least the following important points: usefulness of models and the modeling process; clear *process-view* of a service system as a multi-project resource-constraint system (project = customer to be served); and the sources of operational delays, namely resource- and synchronization-queues.

Fluid Models of Service Networks: Lecture 5 of the ServEng course introduces the fluid approach, which yields the first mathematical models of a service station and a service network. Being intimately related to Empirical models, fluid models provide a bridge from the empirical to the theoretical part of our course. Our Fluid View is taught through several useful families of models: individual-scenario analysis, queue-buildup diagrams, cumulative flows, and spreadsheet models, the latter based on finite-difference approximations of Ordinary-Differential-Equations (ODEs).

The Building Blocks of a Basic Service Station: Starting from Lecture 6, we proceed with a series of lectures on the main building blocks of a stochastic model for a service station. These are: Arrival process, representing customer’s demand; Service process, capturing work per customer that is required from servers; Customers (im)patience, which is what distinguishes service to a human customer from that to, say, a part in a production line. Each building block is introduced, both empirically, mainly via SEESat, and theoretically, via a corresponding mathematical model. (After teaching Arrivals and Services, they are combined to form the workload process.)

Arrivals of Customers: In Lecture 6, the students learn about customers’ arrivals or, in other words, customers’ demand for service. Arrivals to a service system can be studied in several time-scales, each giving rise to a different type of models. In our course, we focus

on operational short and middle-term decision making based on arrival-rate patterns, which is well served by Poisson processes - time-homogeneous and non-homogeneous (the latter modeling daily arrivals in call centers, hospitals and other service systems).

The Service Process: The importance of properly managing the customer-server interaction, or what we call the *service process*, can hardly be underestimated. Lectures 7 and 8 of the ServEng course explore the service process, focusing on its operational attributes: *duration* and *structure* (as opposed to contents or economic worth, for example). Durations are modeled by distributions (empirical, parametric), and structure via Phase-Type models.

Customer (Im)Patience: The subject of customers' (im)patience, e.g. on the phone while waiting to be answered by agent, or in the emergency department while waiting for a nurse or a physician, or on the Internet while waiting for a response, plays a central role in the analysis of service systems. From a practical point of view, (im)patience could lead to customers' *abandonment* which, in our opinion, is perhaps the most important operational performance measure of a call center. (On a personal note, one of the authors (AM) is greatly in debt to the abandonment phenomena - it was, in fact, the original trigger of his data-driven approach to Service Engineering, both research and teaching.) Two important topics that are introduced in the context of (im)patience are Censoring and Hazard Rates. The former because the (im)patience of served customer is left-censored by the actual waiting time; (The actual waiting times of served customers provide lower bounds for their (im)patience.) The latter since hazard rates constitute natural dynamic Models of (im)Patience (following Palm [30], who proposed to use the *hazard rate* of τ as a means for *dynamically* describing (im)patience, or rather customers' irritation as a function of the time of their waiting.)

Stochastic Models of a Basic Service Station: Lectures 10-12 of the course are dedicated to models of a basic service station: with a homogeneous customers' population (as opposed to multi-type) and i.i.d. servers (as opposed to varying-skills). Specifically, Lecture 10 provides a survey of the classical Erlang-C (M/M/n model), and it introduces the 4CallCenters software [11], or 4CC for short - a friendly, effective practical tool for the analysis of basic Markovian queues (Erlang A/B/C and relatives). Lecture 11 reiterates the importance of the abandonment phenomena and presents the basics of Erlang-A (M/M/n + M) - the model which, in practice, is gradually replacing Erlang-C as the queueing engine of workforce-management software. Finally, Lecture 12 covers non-Markovian queues, with general service and inter-arrival distributions (G/G/n).

Operational Regimes and Staffing - QED Queues: A central challenge in the design and management of a service operation in general, and of a call center in particular, is to achieve a balance between *operational efficiency* and *service quality*. This is addressed in

Lecture 13, where we show that, depending the desired balance, several appropriate operational regimes could arise through corresponding staffing rules. Among these, we introduce the ED, QD and QED regimes.

The QED Regime: The QED regime goes hand in hand with square-root staffing: $n = R + \beta\sqrt{R} + o(\sqrt{R})$, where R is the offered-load, and the service grade β is determined by the (limiting) delay probability, say α (and vice versa). The function $\beta(\alpha)$, for Erlang-C, is called the Halfin-Whitt function. For Erlang-A, it is called the Garnett function, and it is computable via a spreadsheet. QED approximations are typically extremely accurate, over a wide range of parameter values, from the very large call centers (1000's of agents) to the moderate-size (few 10's of agents) [19]; this renders QED approximations applicable to small systems, such as those found in healthcare.

Time-Stable Performance of Time-Varying Queues: The staffing rules based on Erlang-B/C/A assume that all system parameters, in particular the offered load, do not vary with time. This is definitely not the case in call centers and most other service systems, which are clearly time-inhomogeneous. Performance analysis of time-inhomogeneity problem can be addressed within the framework developed in Feldman et al. [8], where traditional staffing approaches are rephrased as follows: Given a time-varying arrival process, what is the staffing level (necessarily time-varying) under which operational performance is time-stable? In other words, [8] answers the question of how to time-stabilize the performance of a time-varying system. In the answer, the time-varying Little's Law / offered-load play a central role.

Workforce Management: Hierarchical Operational View: While we teach mainly staffing methods within operational WorkForce Management (WFM), we do emphasize that the spectrum of WFM is far broader. The hierarchy of decisions along the WFM chain goes through the following steps: Forecasting, Staffing, Shifts, Rostering, and Skills-Based Routing. This hierarchy is not unique to call centers (e.g. nurse staffing in hospitals). Except for Rostering, all steps of the hierarchy are covered in either lectures or recitations, with students learning and practicing forecasting, queueing models, shift scheduling (solving an IP), and some SBR principles.

Heterogeneous Customers and Servers: Skills-Based Routing (SBR): Our models have assumed so far iid customers and iid servers. A next step towards the complex reality of service systems is to allow heterogeneity of either customers or servers, or both - which amounts to studying *queueing networks* (recall Section 1.3.) To maintain some level of analytical tractability, one can proceed in one of two ways. Either render complex the topology of the network while maintaining its protocols simple, for example as in Jackson networks.

Or, conversely, assume a simple network topology while allowing complex protocols, as is the case in SBR or Fork-Join networks. In view of time and scope constraints, only one of these models can be covered in ServEng - and we chose SBR: it seemed to be the more natural path when teaching services, and it was consistent with the course's emphasis on call centers. Thus, the last lecture covers SBR. We have been taking a practically-oriented approach, covering parts of the teaching note of Garnett and Mandelbaum [13], and expanding with some industry papers and recent research results when feasible, and as time permits.

5 ServEng Homework and Exams: A Data-Based Approach

In concert with the way the course is taught, both in lectures or recitations, the course assignments and exams are also data-based, when feasible. ServEng students are thus exposed to various real-world data sets, with which they apply theoretical and practical tools that they were taught. The data-based exercises still go hand in hand with ample interesting theoretical homework, so that understanding the material goes beyond the “recipe level”. Homework weighs about 50% towards the final grade, while the rest is determined by a class-exam (3-3.5 hours). In this section we describe both homework and exams.

Data-based homeworks and exams are difficult to create, hence the support of the SEELab is essential here; and, even so, the homeworks must serve the course over a relatively long time, and writing an exam takes many hours and iterations (a joint Instructor+TA effort).

Data analysis tends to require relatively many hours of work from students, some of it not directly rewarding, yet it is very important for the learning experience of ServEng. Hence homeworks are carried out in groups (up to 4 students in earlier generations of the course, and students pairs in recent generations), and their weight in the final grade is substantial - around 50%, as already mentioned. Group assignments also save on feedback and grading resources, which are not in abundance. (High quality feedback is a must for the hard-working students, in order to help them both learn from their mistakes as well as sustain their motivation.) To achieve all the above, the support staff of the ServEng course (TA's, homework graders) must be dedicated and of the highest quality, which has indeed been the case - we return to describing their essential contributions in Section 6.

5.1 Homework Assignments

We now present a complete list of the homework assignments, with accompanying brief descriptions.

5.1.1 List of Assignments

Recitations and assignments, as well as additional related materials, can be downloaded from the ServEng website. Note that one can also download partial solutions for most of the assignments - these were installed when we moved from groups of 4 to pairs, in order to reduce students' workload. (Professors who would like to experiment with teaching ServEng could write to the first author (AM), regarding further information about the homework.)

- 1. On queues.** This introductory assignment requires from students to read some text-book/background materials and present a brief executive summary on “The Future of Queues in Service Systems”.
- 2. Capacity Analysis. Little’s Law.** The assignment contains mostly theoretical questions on Little’s Law; some were in fact motivated by empirical scenarios.
- 3. Empirical Models.** This is the first data-based homework assignment. Data of a U.S. Banking call center, exported via DataMOCCA [33], is prepared as convenient Excel files that students then use. Answering the questions requires the application of lecture material on the Little’s Law, capacity analysis and fluid models in order to .
- 4. Processing Networks.** We ask students to create a model of a process network (See Section 4.3 of Full Version [28].) with which they are familiar. First, an activity precedence diagram should be described, then a resource-based representation should be given and, last, the two representations must be integrated into the combined diagram of the activities and resources. An information flow diagram has been optional in recent years. In addition, students are required to speculate on the Service Engineering of their model, based on their diagrams.
- 5. SEEStat Analysis.** Students download an educational version of SEEStat (the GUI of DataMOCCA), with which they explore part of the U.S. Bank database. They search for answers to several practically-oriented questions, redoing (at the push of a button) some of the exercises that they worked hard over, in the previous Empirical Models homework. We elaborate on this homework in Section 5.1.2 of Full Version [28].
- 6. Fluid Models.** The students address several questions, related to call-center applications, via the fluid approach: they write the proper differential equations, solve them in Excel using the finite-difference method, and finally apply the Solver to optimize some aspect of the operation (e.g. SBR, overtime management). See Section 5.1.2 of Full Version [28] for the detailed coverage of this homework.

- 7. Statistical Analysis of Arrivals.** This homework includes some statistical question (e.g. practical testing of the Poisson hypothesis) and introductory-level forecasting problems.
- 8. Service Processes and Empirical Analysis of Customers' (Im)Patience.** Some questions here are purely theoretical and others, on abandonment behavior, require the interpretation of real-data from call center.
- 9. Gazolco's Call Center.** 4CallCenters (4CC) software [11] is used to analyze various Markovian queueing models (Erlang-C, Erlang-B, Erlang-A and others), which helps develop intuition on some practical call center problems. 4CC is a valuable tool, serving as a personal WFM tool, which students actually use also after graduation.
- 10. Theoretical Analysis of Service Stations in Steady State; Priority Queues.** This homework includes a small case study on pizza delivery, adapted from [9]. It is followed by some queueing-theoretical questions.
- 11. Staffing of Call Centers.** This last homework (in essence, the final project) is based on actual call center data (we have used various call centers across different semesters). It asks the students to apply miscellaneous tools they acquired during ServEng lectures. In one version of the homework, data from three calls centers was used: a small Israeli call center, for applying standard queueing models; a large U.S. call center, that calls for QED analysis; and a medium Italian call center, which went through an interesting process of "hiring and firing".

5.2 The Final Exam

Until several years ago, exams were very comprehensive, requiring a labor-intensive preparatory process. All exams+solutions appear on our website, but they are written in Hebrew. An English-version example of such an exam, with its solution, appears in [6].

The second generation of our exams is of a different form, which is more structured and hence easier to prepare. The exam consists of four questions (for a total of around 50 points = the weight of the exam in the final grade):

- 1. Homework Question:** taken from the homework, and worth about 10-15 points. This question verifies that students were not free-riders when preparing their group homework.

2. **Recitations+Lectures Question:** taken from lecture-notes provided in recitations and lectures, and worth about 10-15 points. This question encourages class attendance, which enhances the learning experience (and which students too often tend to neglect).
3. **Practical Question:** asking the students to actually analyze data (e.g. identifying operational regimes and some of their properties), worth about 15 points. This question in fact tests what students are actually carrying with them after graduation.
4. **Theoretical Question:** that asks students to prove or interpret part of a theoretical fact (e.g. Khintchine-Pollaczek, or Biased Sampling from PASTA, or something similar to a proof from class), worth about 10 points. This question is a pre-requisite for getting one's final-grade up to the level of 90-100. Students who perform well in this question often continue to graduate school (or are already there).

6 Acknowledgements, and a Little More History

The present Service Engineering course at the Technion is the culmination of a long continuous-improvement process. One can view this process as progressing along several fronts:

- *Teaching material:* Most of the material is original, based on research papers of the authors and coauthors, customized teaching notes, graduate theses and students' projects.
- *Recitations and Homework:* These have been developed originally by the authors, and then enhanced, expanded and innovated by the many excellent teaching assistants (TAs) that the course and its students have enjoyed. As already mentioned, homework are central to the learning experience, which is manifested by a weight of about 40-50% of the final grade.
- *Students' Projects:* Undergraduate IE&M student at the Technion must take part in a yearly project, typically carried out during one's last (4th) year of school, with the goal of applying theory to a real-world setting. Similarly, graduate students in the IE&M Operations Research program take a project-course, with the same goal. A full list of projects that the first author has advised appears in [4]. Many of these projects apply tools acquired in the Service Engineering course (mostly to call centers and hospitals) - then excerpts from such projects are often used to demonstrate Service Engineering applications. Other projects constitute pilots that develop into research projects or graduate theses.

For example, Figures 7-10 in Full Version describe the routing process of hospital patients from an emergency department to internal wards. These were first generated

within an undergraduate project at the Rambam hospital in Haifa, then continued as the MSc thesis of Tzeytlin [34], and used as an example for the homework where students are asked to create their own processing network. As another example, Figure 20 in Full Version was first created within a project at a call center in a large Israeli Bank. It has become part of a research paper that is now in the writing.

- *Website*: Maintenance of the course’s website has been the responsibility of the course’s TA. While being a high toll in terms of time and effort, the website has been a central enabler for development and maintenance. Indeed, the website serves as an organized repository of course material, active and archival, for the benefits of students, instructor and TA. Also, through the website, course material is accessible to teachers and research-colleagues world-wide (which is the reason that the material has been developed in English, while the Technion course has been taught in Hebrew).

Course TAs: The Service Engineering course owes much of its success to its excellent TAs. These have been carefully chosen and then well-groomed for the job (for example, by re-attending lectures as a future TA, and learning “trade secrets” from the present TA). Our TAs have all been top graduate students, doing research directly related to some course subjects, and willing to undertake the challenging job of a ServEng TA. This entails first the routine yet time-consuming chores of conducting recitations, maintaining the course website, having office hours (which are usually attended due to the challenging homework) and, all in all, help the course maintain a service-level that a Service Engineering course is worthy of. In addition, there is also the creative part of a TA work (which distinguishes the excellent from the merely very-good) - that of generating new material for recitations and homework, and writing questions for exams, with an emphasis given to incorporating the TA’s own research into course material. One of the authors (S.Z.) was the first TA for the course in its present form. Then, in chronological order, we would like to acknowledge the significant contributions of Itay Gurvich, Gennady Shaikhet, Shimrit Maman and Galit Yom-Tov.

Students’ Contribution: There is a long list of (mostly graduate) students who have contributed to the course material development (sometimes while serving as homework graders). Among these, we would like to especially acknowledge Sivan Aldor-Noiman, Yonit Baron, Izik Cohen, Ofer Garnett, Zohar Feldman, Eva Ishay, Polina Khudyakov, Pablo Liberman, Yariv Marmor, Michael Reich, Luba Rosenshmidt, Arik Senderovich, Yulia Tzeytlin and Asaf Zviran.

The SEELab: Permanent and temporary members of the SEELab [32] have supported the course in various ways, mostly related to data and SEEStat. Here we would like to acknowledge Valery Trofimov, Ella Nadjarov, Igor Gavako, Katya Kutsy and Arik Senderovich.

Research Partners: When teaching, especially a young subject, it is useful for teachers to reflect and students to study the subject’s evolution. This is also an opportunity to acknowledge those who contributed to that evolution. An example is the overhead in Figure 1, taken from a lecture on Queues with Impatient Customers: it describes the “Modeling History” of such queues, as inspired by Call Centers; it also lists the models and their developers, who have been our coauthors and partners, and to whom we are highly grateful. Names of graduate students are highlighted in the slide, this in order to stimulate the curiosity and appetite of course students, towards projects and graduate studies in the field of Service Engineering.

Administration Support: Special thanks is due to the current dean of the IE&M Faculty at the Technion, Professor Boaz Golany, who followed closely the development of the course and spared no resources to nurture it.

Mini-Courses: Mini-versions of the course have been taught at several universities: Wharton (hosted by Larry Brown, Noah Gans and Morris Cohen), Columbia GSB (hosted by Awi Federgruen and Ward Whitt), Stanford GSB (Mike Harrison and Sunil Kumar) and INSEAD (hosted by Zeynep Aksin). These mini-courses gave us the opportunity to share and learn from our hosts and their students - their generosity, along all dimensions, is greatly appreciated.

“Customers” or “Products”: Is the process of teaching Service Engineering a production process (contributing to the production of Service Engineers) or a service process (serving students)? This is a question worthwhile discussing in a first lecture of a Service Engineering course, but we shall not pursue it here. We shall only say that, as either “Service-Providers” or “Producers”, in short Teachers, we are grateful to the many students of Service Engineering, who have learned and labored through the course material and homework: undergraduates, graduates, postdocs, researchers, practitioners and colleagues. The Technion, and the above mentioned hosting institutions for the mini-courses, are blessed with students of high quality and drive, which makes the teaching experience (more often than not), and hopefully the learning experience as well, challenging, enjoyable and rewarding. Indeed, learning that a course-graduate keeps the lecture notes handy on an office shelf, or having the course be the trigger for a student’s successful academic career, is all that a teacher can hope for.

Figure 1: **Acknowledgement to Partners, Coauthors and Students**

**Call Centers = Q's w/ Impatient Customers
14 Years History, or "A Modelling Gallery"**

1. Kella, Meilijson: Practice \Rightarrow Abandonment important
2. Shimkin, Zohar: No data \Rightarrow Rational patience in [Equilibrium](#)
3. Carmon, Zakay: Cost of waiting \Rightarrow [Psychological](#) models
4. Garnett, Reiman; Zeltyn: Palm/Erlang-A to replace Erlang-C/B as the standard [Steady-state](#) model
5. Massey, Reiman, Rider, Stolyar: Predictable variability \Rightarrow [Fluid](#) models, [Diffusion](#) refinements
6. Ritov; Sakov, Zeltyn: Finally Data \Rightarrow [Empirical](#) models
7. Brown, Gans, Haipeng, Zhao: [Statistics](#) \Rightarrow Queueing Science
8. Atar, Reiman, Shaikhet: Skills-based routing \Rightarrow [Control](#) models
9. Nakibly, Meilijson, Pollatchek: Prediction of waiting \Rightarrow [Online Models and Real-Time Simulation](#)
10. Garnett: Practice \Rightarrow [4CallCenters.com](#)
11. Zeltyn: Queueing Science \Rightarrow [Empirically-Based Theory](#)
12. Borst, Reiman; Zeltyn: [Dimensioning](#) M/M/N+G
13. Kaspi, Ramanan: [Measure-Valued](#) models and approximations
14. Jennings; Feldman, Massey, Whitt: [Time-stable performance](#) (ISA)

References

- [1] Borst S., Mandelbaum A., and Reiman M. (2004), Dimensioning large call centers, *Operations Research*, 52(1), 17-34. [1.4.2](#)
- [2] Brown L.D., Gans N., Mandelbaum A., Sakov A., Shen H., Zeltyn S. and Zhao L. (2005) Statistical analysis of a telephone call center: a queueing science perspective. *Journal of the American Statistical Association (JASA)*, 100(469), 36-50. [3](#), [4](#)

- [3] Corbett, C.J., van Wassehove, L.N. (1993) The natural drift: what happened to Operations Research?, *Operations Research*, 41, 625-640. [1.5.3](#)
- [4] CV of A.M., available at http://iew3.technion.ac.il/serveng/References/AviM_cv_08.pdf. [6](#)
- [5] Erlang A.K. (1948) On the rational determination of the number of circuits. In *The life and works of A.K.Erlang*. Brockmeyer E., Halstrom H.L. and Jensen A., eds. Copenhagen: The Copenhagen Telephone Company. [1.4.1](#)
- [6] Example of a ServEng exam. Available at http://iew3.technion.ac.il/serveng/Lectures/Exams/Moed_A_2004_Eng.pdf. [5.2](#)
- [7] Feldman Z. (2008) Optimal Staffing of Systems with Skills-Based-Routing. Technion M.Sc. thesis. Available at http://iew3.technion.ac.il/serveng/References/Zohar_Thesis.pdf. [1.3.2](#)
- [8] Feldman Z., Mandelbaum A., Massey W. and Whitt W. (2008) Staffing of time-varying queues to achieve time-stable performance. *Management Science*, 54(2), 324-338. [4.1](#)
- [9] Fitzsimmons J. and Fitzsimmons M. (2004) Service Management: Operations, Strategy, Information Technology McGraw Hill, 4th Edition. [1.1](#), [4.1](#), [5.1.1](#)
- [10] Flanders S. (1980) Modeling court delay. *Law & Policy Quarterly*, 2, 305320. [4.1](#)
- [11] 4CallCenters Software (2002). Available at <http://iew3.technion.ac.il/serveng/4CallCenters/Downloads.htm>. [4.1](#), [5.1.1](#)
- [12] Frei F. X., Harker P. T., Hunter L. W. (1998) Innovation in Retail Banking. Report, Wharton's Financial Institutions Center. Available at <http://iew3.technion.ac.il/serveng/Lectures/Retail.pdf>, or <http://fic.wharton.upenn.edu/fic/papers/97/9748.pdf>. [1.1](#), [4.1](#)
- [13] Garnett O. and Mandelbaum A. (2000) An Introduction to Skills-Based Routing and its Operational Complexities. Teaching note, Technion, Israel. Available at <http://iew3.technion.ac.il/serveng2004/Lectures/SBR.pdf>. [4.1](#)
- [14] Garnett O., Mandelbaum A. and Reiman M. (2002) Designing a telephone call-center with impatient customers. *Manufacturing and Service Operations Management*, 4, 208-227. [1.4.2](#)

- [15] Halfin S. and Whitt W. (1981) Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29, 567-588. [1.4.2](#)
- [16] Hall R.W. (1991) *Queueing Methods for Services and Manufacturing*, Englewood Cliffs, New Jersey, USA: Prentice Hall. [1.4.1](#), [4.1](#)
- [17] R.W. Hall (Editor) (2007) *Patient Flow: Reducing Delay in Healthcare Delivery* (International Series in Operations Research and Management Science). [1.4.1](#)
- [18] Herman, R. (1992) Technology, human interaction, and complexity: reflection on vehicular traffic science, *Operations Research*, 40(2), 199-212. [1.5.3](#)
- [19] Janssen A.J.E.M. , van Leeuwen J.S.H. , Zwart B. (2008) Refining square root safety staffing by expanding Erlang C, to appear in *Operations Research*. [1.4.2](#), [4.1](#)
- [20] Jennings O. and de Vericourt F. (2007) Nurse-to-patient ratios in hospital staffing: a queueing perspective. Working Paper, Duke University. [1.4.2](#)
- [21] Khudyakov P. (2006) *Designing a Call Center with an IVR (Interactive Voice Response)*. M.Sc. Thesis, Technion. Available at http://iew3.technion.ac.il/serveng/References/thesis_polyna.pdf [1.4.2](#)
- [22] Little J.D.C. (1961) A Proof of the Queueing Formula $L = \lambda W$, *Operations Research*, 9, 383-387. [4.1](#)
- [23] Lovelock. C.G. (1992) *Managing Services: Marketing, Operations and Human Resources*, Prentice-Hall. [4.1](#)
- [24] Mandelbaum A., Massey W.A. and Reiman M. (1998) Strong Approximations for Markovian Service Networks. *Queueing Systems: Theory and Applications (QUESTA)*, 30, 149-201. [1.4.2](#)
- [25] Mandelbaum A., Sakov A. and Zeltyn S. (2000) *Empirical Analysis of a Call Center*. Technical report, Technion. [3](#)
- [26] Mandelbaum A. and Zeltyn S. (1998) Estimating characteristics of queueing networks using transactional data. *Queueing Systems: Theory and Applications (QUESTA)*, 29, 75-127. [3](#)
- [27] Mandelbaum A. and Zeltyn S. (2007) Service engineering in action: the Palm/Erlang-A queue, with applications to call centers. In: Spath D., Fähnrich, K.-P. (Eds.), *Advances in Services Innovations*, 17-48, Springer-Verlag. [4](#)

- [28] Mandelbaum A. and Zeltyn S. (2009) Service Engineering: Data-Based Course Development and Teaching. Full Version. A full version of the present document, under review for *INFORMS Transactions on Education*, Special Issue on “Teaching Service and Retail Operations Management”. *, 1, 1.3.2, 1.4.1, 1.5.4, 4, 4.1, 5.1.1
- [29] Newell G.F. (1982) Application of Queueing Theory, Chapman and Hall. 1.4.1
- [30] Palm C. (1957) Research on telephone traffic carried by full availability groups. Tele, vol.1, 107 pp. (English translation of results first published in 1946 in Swedish in the same journal, which was then entitled *Tekniska Meddelanden fran Kungl. Telegrafstyrelsen*.) 1.4.1, 4.1
- [31] “Service Engineering” course website, Technion, <http://iew3.technion.ac.il/serveng>. (document), 1.1, 2, 4.1
- [32] SEE: Website of the Technion’s “Service Enterprise Engineering” Research Center, <http://ie.technion.ac.il/Labs/Serveng/>. (document), 3.1, 6
- [33] Trofimov V., P.D. Feigin, Mandelbaum A., Ishay E., and Nadjharov E. (2006) DATA MModel for Call Center Analysis: Model Description and Introduction to User Interface. Technion, Israeli Institute of Technology, Technical Report. Available at <http://ie.technion.ac.il/Labs/Serveng>. 3.1, 5.1.1
- [34] Tseytlin Y. (2009) Queueing Systems with Heterogeneous Servers: On Fair Routing of Patients in Emergency Departments. Technion M.Sc. Thesis, April 2009. Available at <http://iew3.technion.ac.il/serveng/References/thesis-yulia.pdf>. 6
- [35] U.S. Bank SEELab Report (2006) DataMOCCA: DATA MModel for Call Center Analysis - The Call Center of “US Bank (120 pages). 3
- [36] Whitt W. Research site. <http://www.columbia.edu/~ww2040/allpapers.html> 1.4.2, 1.4.3
- [37] Yom-Tov Galit (2007) Queues in Hospitals: Semi-Open Queueing Networks in the QED Regime. Ph.D. Research Proposal, Technion. Available at http://iew3.technion.ac.il/serveng/References/proposal_Galit.pdf 1.4.2
- [38] Zeltyn S. and Mandelbaum A. (2005) Call centers with impatient customers: many-server asymptotics of the M/M/n+G queue. *Queueing Systems: Theory and Applications (QUESTA)*, 51, 361-402. 1.4.2

- [39] Zviran A. (2008) Fork-Join Networks in Heavy Traffic: Diffusion Approximations and Control. M.Sc. Research Proposal, Technion. Available at http://iew3.technion.ac.il/serveng/References/Asaf_research_proposal.pdf

1.4.3