INFORMS Journal on Applied Analytics

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Service-Delivery Modeling and Optimization

Yixin Diao, Aliza Heching, David Northcutt, Rodney Wallace

Interfaces
Vol. 45, No. 3, May–June 2015, pp. 243–259
ISSN 0092-2102 (print) | ISSN 1526-551X (online)
http://dx.doi.org/10.1287/inte.2014.0783
© 2015 INFORMS

Service-Delivery Modeling and Optimization

Yixin Diao, Aliza Heching
IBM T.J. Watson Research Center, Yorktown Heights, New York 10598
{diao@us.ibm.com, ahechi@us.ibm.com}

David Northcutt, Rodney Wallace
IBM Global Technology Services, Somers, New York 10589
{dnorthcutt@acm.org, rodney.wallace@us.ibm.com}

Service-delivery modeling and optimization is a complex problem involving multiple service levels, skill sets, request classes, and service times. In this paper, we describe a comprehensive and scalable end-to-end analytical methodology that we developed and implemented at a global information technology service delivery provider. This methodology provides predictive insight and prescriptive solutions to the problem of staffing service-delivery units in this complex environment. Our solution has been deployed globally at more than 640 service-delivery units and has yielded more than $52 million in cost savings and cost avoidance to date.

Keywords: service system; simulation model; optimization.
History: This paper was refereed. Published online in *Articles in Advance* April 3, 2015.

In recent years, the information technology (IT) service industry has faced continual pressure to improve the quality of the services that it delivers to customers while simultaneously reducing delivery costs. These apparently conflicting objectives have led the industry to explore innovative methods for managing the businesses within it. To that end, service providers are focusing on and measuring their internal processes, the skills of their people, and their organizational structure. However, the inherently labor-intensive nature of the IT service industry demands complicated trade-offs to ensure that customer requests are satisfied within contractually specified service-quality targets, and correct staffing decisions are made on matters such as agent skills, cross-training, and temporary labor. In this paper, we describe a process-based business analytics methodology that integrates key components of the staffing decision-making process into a complete and scalable prescriptive services solution. It includes an optimization framework that supports staffing decision making in complex service-delivery systems; this framework considers the relationships among customer workload, contractual service-level constraints, agent skills, and shift schedules.

We consider a service-delivery system in the context of global IT service delivery. Global delivery refers to a model for delivering IT services in which the service provider services global customers from either on- or off-shore locations. A customer may have multiple IT needs; examples include managing a network, supporting a database, requiring backup and restore services, and building and configuring new servers. After reviewing the menu of IT services that the service provider offers, a customer contracts for one or more services. The infrastructure supporting the services (e.g., servers, networks, application, business processes) may be owned by the customer and located on a customer site or may be owned and maintained on provider sites on behalf of the customer. Services are provided from one of the service provider's globally located delivery centers. Arriving customer requests for service (i.e., service requests) are assigned to a service-delivery location; agents at this location are responsible for responding to the service requests. Although the agents in these locations respond to the service requests, they usually do not directly interact with the end customers.

We now describe the service-request life cycle in the service-delivery process (see Figure 1). A service request typically arrives via one of the following sources:

- Following service interruption, the end user (customer) reports an issue via a Web-based ticketing system.

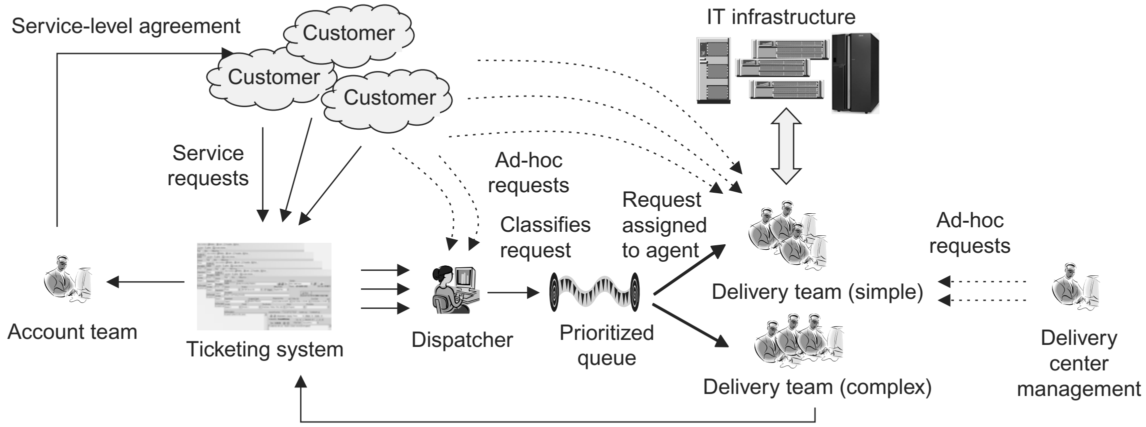

Figure 1: The service-delivery process shows the life cycle of a customer service request. A dispatcher monitors arriving requests, prioritizes requests based on business impact, and assigns requests to delivery teams based on required skills within these teams.

- Help desk personnel who cannot resolve a customer inquiry create a service request for second-level support.
- One of the service provider's agents, who proactively monitor customer systems, identifies a problem and creates a service request.
- An alert monitoring system, which monitors customer systems for indications of system failure, identifies a failure and automatically creates a service request.

Details of each service request, including customer name, creation date and time, request severity, and problem description are documented in a ticket. In addition to formal service requests, ad-hoc requests for service, such as ad hoc reports for the customer, may occur.

Arriving service requests are classified according to attributes such as required skills and level of priority. The requests are then routed to a service-delivery unit (SDU) based on a fixed mapping that considers aspects such as technology area and customer assignment. An SDU comprises one or more service-delivery teams; each team has a group of service agents who have common skills and are capable of responding to similar service requests. Upon the request's arrival at the SDU, a dispatcher reviews and prioritizes it and assigns it to an agent in one of the teams based on factors such as request severity, agent skill, and agent availability. Dispatching is a critical component in service operation management to ensure proper request handling. Typically, the request priority is determined based on the request severity that was specified at the request creation time. Request severity reflects the urgency of the request, and is associated with contractual service-level agreements (SLAs) that specify target resolution times and attainment levels. The agent services his (her) assigned requests in order of request priority. An agent may manage multiple requests simultaneously (i.e., multitask), batch process similar requests, place requests on hold (e.g., to wait for additional customer input), help out another team during heavy load periods, or preempt requests in service because of the arrival of higher-priority requests. An agent may also need to service ad-hoc requests from management, such as auditing and group training. The dispatcher manages all this activity based on the volume of workload that the SDU must manage.

The last step in the service-delivery process is service performance measurement and reporting, which must be performed at various stages and levels in the service-delivery process. For example, request response time (i.e., time from the creation of a request until an agent first begins to service it), request resolution time, and backlog level (i.e., total number of outstanding service requests) are typically reported at the service-delivery team and SDU levels as performance-management and quality indicators. These measures are also reported at the customer level because service-level management calculates SLA attainment on a customer-by-customer basis.

The IT service-delivery environment is complex. An SDU typically supports a large variety of requests arriving from multiple sources (e.g., email, Web, instant message, ticketing systems). It also supports multiple customers, each with different contractually dictated times for response and resolution. Requests usually have an associated priority that relates to the business impact of the particular request and is determined by the customer. Contractual SLAs are associated with the requests. These agreements specify the target response and resolution times (which can vary from minutes to hours to days) for each request; target response times vary by customer, request type, and request priority. SLAs are usually structured in such a way that rather than measuring the performance of each request relative to the contractual SLA, a given percentage of requests in a given request class and arriving within a contractually specified period must meet the target response time. Responding to different requests may require different skills; for example, a password reset requires less skill than a new server build.

Within this complex environment, we were tasked with identifying optimal staffing levels for a service provider with globally distributed SDUs. Our objective was to identify optimal staffing levels for each globally distributed SDU and each agent team that comprise the various SDUs. Given the dynamic nature of the delivery environment, the data-driven solution that we developed would also be used to explore the impact of changes in the delivery environment that might potentially impact required staffing levels, for example, changes in incoming workload, contractual SLAs, and agent skills. Our solution would also consider the benefits of additional training for agents. We would provide agents with skills in other areas that match the complexity of their current skills, and with skills that are more complex than they presently have, enabling them to perform more complex tasks.

To approach this problem, we considered off-the-shelf solutions including both analytical (optimization-based) and simulation-based solutions. We reviewed a number of workforce management (WFM) tools and determined that no WFM tool would meet all our needs because of the complex nature of the delivery environment. Therefore, we quickly decided that we must create our own solution rather than use an off-the-shelf WFM tool. Further, we found that only a simulation tool would allow us to fully model the complexities of the service-delivery environment. By using a commercial discrete-event simulation tool that allowed us full control via its programmatic interface, we were able to take advantage of the power that the simulation package offered, but still customize the tool to accommodate our unique features, such as a custom dispatching process. The programming interface would also allow us to make changes going forward and not be tied to feature availability in a commercial product.

We developed a suite of mathematical tools and models; our objective was to determine the required number of agents and the best mix of skills in each SDU to optimize service-delivery performance. The complex delivery environment demanded a solution customizable to the needs of many SDUs, yet scalable across hundreds of SDUs. The solution had to be simple to use so that entry-level analysts could deploy it, and it had to be accurate enough to instill confidence.

At a high level, we can summarize the innovation and creativity in our work in three key areas, as follows.

- Integration of multiple analytical techniques to develop an end-to-end business solution.

(1) Descriptive: We use probability and statistics and text-mining techniques to analyze historical workload, work activity data to derive probability distributions and temporal patterns, and statistical methods to identify outliers as part of our data cleaning. Work activity data, which we describe in more detail in the *Data Collection* section, refers to the detailed, time-stamped data on all activities the agents perform during their shifts.

(2) Predictive: We use simulation to model complex relationships among workload type and volume, staffing skill and schedule, and contractual SLAs.

(3) Prescriptive: We use simulation-optimization to generate optimal staffing levels by shift and agent skill level. Differentials in agent costs (e.g., because of skill levels) and costs of SLA penalty violations are considered, as are other contractual restrictions

that may be present (e.g., minimum number of agents per period).

- Development of new or nonstandard methodologies to address unique challenges in service delivery.

(1) Workload analysis: A workload pattern analysis method detects and eliminates fake service bursts because of data recording issues.

(2) Simulation modeling: A heuristic algorithm assigns the workload to agents with specialized skills, and a flexible queue priority policy considers both job processing times and job priorities.

(3) Optimization: A nonstandard method for measuring confidence bounds is aligned with system performance metrics, a two-stage optimization procedure minimizes disruptions in current team operations with no impact on the optimal cost, and several features reduce the simulation run time using warm start, additional constraints, and dynamic feedback.

- Large-scale and streamlined application of data-driven analytics to enable fact-based decision making in a business environment.

(1) Process: We employ an end-to-end deployment process consisting of five major steps with a span of 16 weeks and enforced consistent process execution through semi-automatic project status tracking.

(2) Tooling: Data templates, a data portal and timing tool, a data preprocessor, and a modeling and optimization engine make the process scalable.

We organized the remainder of this paper as follows. The *Optimal Delivery Staffing* section describes our modeling methodology for service-delivery system modeling and optimization. The *Scaling Deployment* section discusses our work to scale the solution deployment. The *Business Results* section summarizes the impact generated from this work. The *Lessons Learned and Future Work* section highlights our modeling and deployment lessons and discusses future directions. Our conclusions are contained in the *Conclusions* section.

Optimal Delivery Staffing

Related Work

The problem that we describe falls into the area of optimal staffing with skills-based routing (SBR); customer requests arrive with specific skill requirements, and are serviced by agents with corresponding skills. The SBR problem is known to be analytically complex with limited theoretical results. Gans et al. (2003) and Aksin et al. (2007) provide detailed surveys of the analytical approaches that have been undertaken. The most common approaches are to simplify either the topology of the network or routing schemes; however, none of these approaches is desirable in our environment where both the network and the routing schemes are complex and the service providers are seeking practical solutions, not conceptual guidance.

An alternative solution to the SBR problem is the simulation-based approach. Simulation derives suggested solutions after considering the complexities of the real-world system, such as the nonstationarities in the arrival rates and the interactions between decisions made in different periods. A common model is to adopt a two-stage approach wherein optimization is used to generate a starting solution and simulation is used to evaluate real-system feasibility (e.g., service-level attainment) of this analytical model-suggested solution. Atlason et al. (2008) consider a multiperiod problem of determining optimal staffing levels while meeting service-level requirements. They solve a sample average approximation of the problem using a simulation-based analytic center cutting-plane method and assuming that the service-level functions are pseudoconcave. Cezik and L'Ecuyer (2008) extend this approach by applying it to large problem instances and developing heuristic methods to handle the numerical challenges that arise. Feldman and Mandelbaum (2010) use stochastic approximation to determine optimal staffing levels, assuming that the service-level functions are convex in the staffing levels. They consider two model formulations, one in which the service levels are strict constraints and the second in which the service levels are entered as costs in the objective function, and use simulation to evaluate service-level attainment. Robbins and Harrison (2008) consider a two-stage approach for determining optimal staffing levels in a call center environment. In the first stage, they solve for the staffing levels by using per-period attainment as an approximation for the true service-level attainment. In the second stage, the simulation is used to evaluate true system performance and service-level attainment. Bouzada (2009) describes the use of simulation to determine optimal staffing levels in a call center environment. The author also reports on sensitivity of the service-level attainment abandonment rate to model parameters such as changes in handling time distribution, call volume, or SLA constraints. Anerousis et al. (2010) use simulation to study how various operational scenarios affect optimal staffing in a service-delivery organization, considering a diverse skill base, the presence of service-level objectives, and incoming work with varying levels of complexity.

High-Level Approach

To solve the optimal delivery staffing problem, we considered both analytical methods and simulation modeling. The recommended staffing levels would be used to drive operational decisions. As such, it was critical to develop a model that accurately reflected the complexities of the business environment, including factors such as the manner in which the agents prioritize service requests and the nature of the SLAs. We found that capturing all these real-world complexities using an analytical model would be challenging. We therefore decided to develop a discrete-event simulation model (DESM) to accurately capture the critical system complexities (Diao et al. 2011). The DESM is designed to support decision making at the SDU level after the service provider has made decisions regarding the assignment of customers to SDUs.

We developed the DESM based upon extensive observation of the practices followed by the globally distributed SDUs. Each modeler spent several weeks at a number of globally distributed delivery centers to observe different SDUs and learn the customer service-request life cycle. Visiting these global sites provided the modelers with insights into process differences between SDUs; these differences may be attributed to factors such as cultural differences, legal restrictions, or maturity of location.

The DESM models the arrival of requests to the SDU, prioritization of those requests, and assignment of requests to agent teams according to the assignment policies that the agent teams follow. Request service times are assigned according to the request class. Rules specify how requests that arrive during off-shift hours (i.e., when no agents are working) should be serviced as well as the service of requests that are in process at the time an agent completes his (her) shift. Two (conflicting) performance metrics measured are (1) agent utilization, and (2) service-level attainment.

Data Collection

Three detailed categories of data were required as input to the DESM: (1) workload data, (2) work activity data, and (3) demographic data. The workload data are used to build arrival-rate distributions for each class of request serviced by the agent teams; the work activity data are used to build the service-time distributions associated with each class of request; demographic data provide information about the agent teams, including the skills of the agents, number of shifts, shift hours, and contractual obligations regarding agent coverage. We now describe in detail each data category and how the DESM uses the data.

Workload data refer to historical data of all incoming requests by request class, where a request class is defined as a group of requests with the same priority, require identical skills to service, and are measured against the same SLA. For each request class, we collect six months of historical workload data. We conduct extensive statistical analysis of the data to identify and eliminate outliers and identify any temporal patterns or shifts over time. The historical data are used to estimate the arrival-rate distributions for each request class. Arrival-rate distributions vary by request class. For example, change requests typically experience higher volume during the (customer) nighttime hours and on weekends. The volume of alerts is often highly related to the usage of the customer systems; an online retailer may see higher volumes during the lunchtime and early evening hours. Another type of workload is so-called scheduled workload—requests that must be serviced on specific days at specific times. For this scheduled workload, we obtain the schedule of the work. In some cases, the request must be serviced at a specific time (e.g., every Monday at 9 AM, check if system X is functioning). In other cases, the schedule specifies that a check must be performed within a specific time interval (e.g., every Monday between 1 PM and 4 PM, perform a health check on system ABC).

Figure 2 displays plots of hourly volume of requests for each hour of the week for two request classes. The x-axis represents the hours of the week, beginning at midnight on Sunday and ending at 11 PM Sunday. The y-axis represents the volume of requests per hour. The top plot (A) displays a pattern in which an increase in volume of requests is observed each

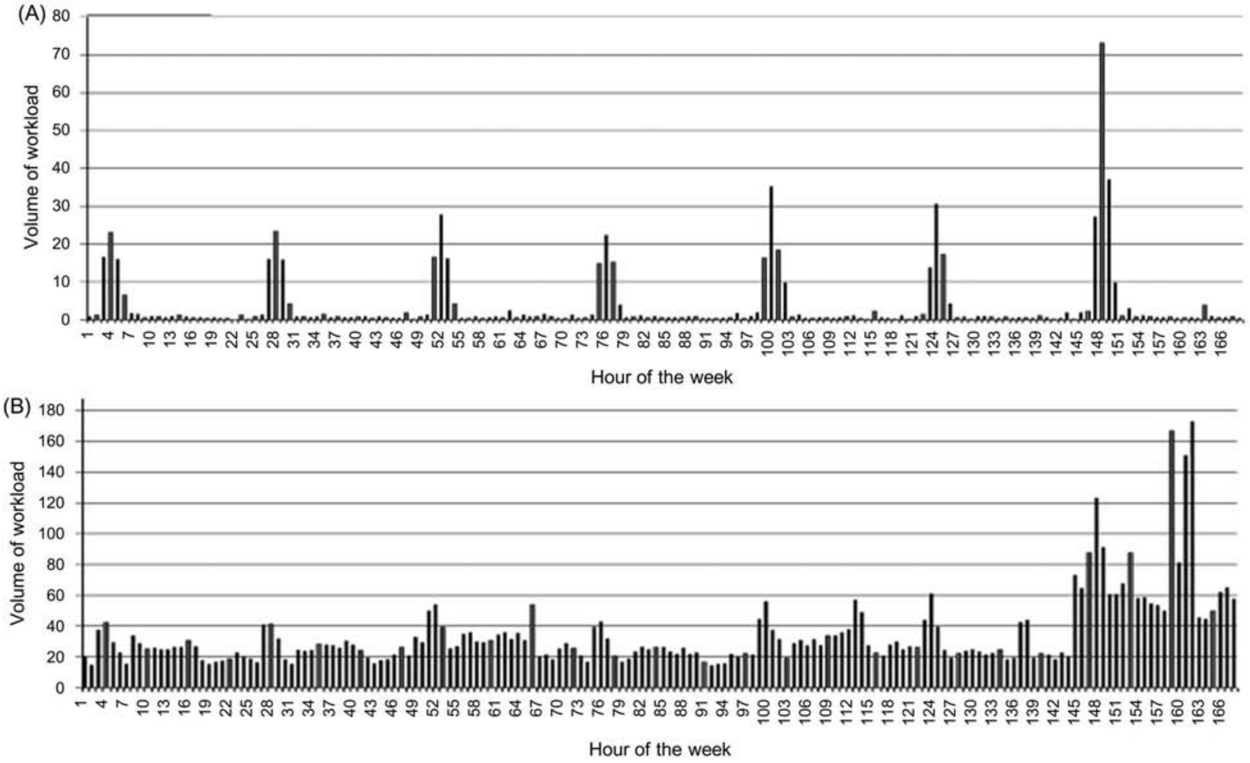

Figure 2: The plot shows the weekly volume of requests by the hours of the week. Plot A exhibits a demand pattern where volume increases every evening between 11 PM and 2 AM; Plot B exhibits a demand pattern where volume increases during the weekend.

night between 11 PM and 2 AM. The bottom plot (B) displays the workload pattern for a request class in which the volume of requests increases on Saturday and Sunday.

The workload data are also used to derive templates for skills required to respond to different requests. Specifically, we use the historical data to measure the depth and breadth of skills required for each request class. We also use these data to build the mapping of attributes of requests to required agent skills.

Work activity data are detailed data that are collected regarding each activity performed by each agent, including the exact time the agent starts and stops the activity. The data include the request type, customer, priority, and complexity. These data are collected through a tool installed on each agent's workstation. The agent records each time he (she) starts, pauses, or completes an activity. An activity may be started and stopped multiple times if the agent is interrupted while servicing a request. Agents may also indicate that they are multitasking. For example, an agent who is at a team meeting and also working on a computer to resolve a customer request would record two concurrent activities in the tool (team meeting and customer request); these activities would have overlapping start and complete service times.

The work activity data, which rely on the agent for data collection (agents record the start and stop times for each activity in this data set), are subject to human input error. For example, we often observe that agents forget to stop activities, such as before taking a lunch break or leaving at the end of a shift. In these cases, we also observe inaccurately long service times. Agents may also mistakenly create activities after which they would immediately stop the activity in the tool. In this case, we would observe inaccurately short service times associated with the activity. We perform statistical analysis of the data collected to identify and eliminate outliers. The remaining data are used to estimate the service-time distribution for each request class. We distinguish between service time and time in system because time in system may include time that the request is waiting (e.g., waiting for additional information or interrupted by other requests).

In some cases, insufficient volume of data is available to produce statistically valid estimates of service-time distributions for each request class. In such cases, we use business knowledge to identify similar request classes whose data can be aggregated to compute statistically valid estimates of service-time distributions.

Figure 3 provides a visual depiction of how the work activity data are used to estimate the service-time distributions for each request class. In line with the findings of Brown et al. (2005), service times are modeled as a lognormal random variable.

We analyze the service times across agent teams (for any given request class) and identify variations in the parameters of the distributions of the service times. Table 1 provides one comparison across four agent teams. As part of our analysis, we investigate these differences in service times and reveal underlying explanations, such as differences in contractual obligations, supporting systems, or operating business environments.

The demographic data include information about the customers supported by each SDU, the number of agents in each SDU and their skills, the working hours, and the defined shifts supported by each SDU. Information about the customers includes the full list of request types supported for each customer, terms of the SLA, customer business hours, and customer time zone relative to the SDU time zone. We also collect information regarding any contractual constraints, such as minimum number of available agents during various hours of the day (irrespective of the volume of workload) or restrictions on the agents who may service an account (e.g., because of privacy issues).

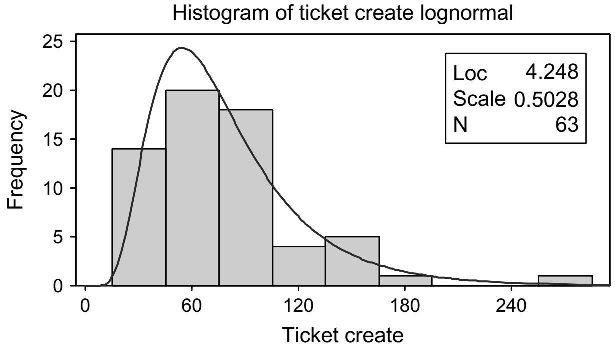

Figure 3: We use work activity data to estimate the distribution of the service for each class of request performed in the service-delivery center. A lognormal distribution fits the data well.

Agent team	No. of Agents	Volume requests/ day/shift	Mean	Median	Standard deviation
AT-1	33	160	14.16	7.65	12.50
AT-2	15	123	14.56	11.9	11.21
AT-3	21	43	12.11	8.5	12.61
AT-4	21	66	9.76	6.8	8.89

Table 1: Service-time distributions across agent teams vary within the same request class. This may be explained by differences in contractual requirements or business processes, or opportunities for process improvement and best-practices sharing.

Information gathered about the configuration of the SDU includes the SDU working hours (e.g., 24/7 or nine-to-five), number of shifts, and shift specifications. Finally, we gather information about the agents in the SDU, including the role of each agent, percentage availability (e.g., part-time employee versus full-time employee), and breadth and depth of skills. We also record the current shift to which the agent is assigned.

Simulation Model

We use a fully parameterized and highly flexible approach to designing the DESM. This flexible design allows us to tune the model to replicate the processes followed in each SDU including, for example, dispatching, queuing, and service processes.

The DESM, which we developed on the AnyLogic® Technologies simulation modeling platform, simulates the service-request life cycle for each request serviced by each SDU. The objective of the DESM is to determine the optimal number of staff members with each of the required skill levels for each shift to minimize the total cost of delivery. We measure cost as the total cost of labor, where more-skilled agents have a higher associated labor cost. A number of constraints restrict the optimal solution. The first set of constraints involves any contractual constraints, such as the minimum number of agents available during specific hours of the day or restrictions on the agents who may respond to specific types of requests. These constraints are SDU specific. The second set of constraints relates to the SLAs and constrains the suggested staffing requirements per shift to ensure that SLAs are met. SLA attainment is only measured against requests that have completed service. To ensure that a backlog of requests was not accumulating in the system, we introduced a third set of constraints that measured the volume of backlog and changes in backlog over time. (We quickly found that systems with high volumes of backlog took a long time to simulate because of the need to prioritize the large number of requests in the queue. We therefore were able to use the time to evaluate a scenario as a proxy for high backlog and lack of feasibility of the solution under evaluation.)

Figure 4 depicts the main modules of the DESM that correspond to key components of service delivery in the SDU. The arrival module specifies the number of request classes and their associated arrival processes. Each simulated arrival is tagged with an arrival time (i.e., time when the request was generated), which is used to measure service-level attainment, request class (associated with additional attributes such as its request type, customer, complexity, priority, and required skills), and service time (sampled from the service-time distribution associated with the request class).

The dispatching module assigns requests to the appropriate agent team. We developed a number of routines for this module, each corresponding to a dispatching policy; these routines are enabled by appropriately specifying the parameters that control the dispatching module. This module includes rules for how to service incoming requests during off-shift hours and whether requests are held in a single central queue or routed to agent team-specific queues. The queuing module specifies the policies to determine (1) how requests are managed in the queue(s) before they are assigned to an agent for service, including prioritization between requests, and (2) which agent team should respond to a request when multiple agent teams have the skills to respond to it. The service module describes the service policies, including how requests that are in the middle of being serviced at the end of a shift should be handled and how agents should respond when requests are assigned during off-shift hours.

Finally, data gathered in the exit module are used to compute system performance. We measure service-level attainment for each request class and agent utilization for each agent team. We also monitor system backlog to ensure that the volume of requests in the system (for each request class or across all request classes for each agent team and SDU) is not increasing over time.

We measure contractual SLA performance on a monthly basis, and therefore simulate SDU performance over one-month intervals. For each scenario,

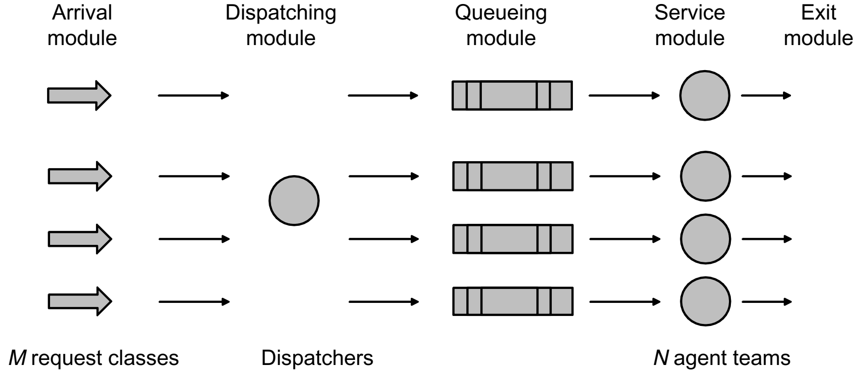

Figure 4: The DESM is comprised of a number of modules that represent the key components of service delivery in an SDU. The functionality of each module is modeled after the behavior that we observe in the real world.

we first simulate a sufficient number of periods to eliminate any effect of initial transient periods. We then simulate a sufficient number of replications of each scenario to ensure tight confidence bounds on the estimates of the performance metrics.

Solution Approach

The DESM is used to measure system performance assuming a specified number of agents of each skill level during each shift. The service provider's objective is to identify the optimal number of agents of each skill level to assign to each shift. Toward this end, we adopt a two-stage framework to determine the optimal staffing levels: In the first stage, we utilize simulation-based optimization, a guided method for selecting and evaluating alternative solutions, to identify a minimum-cost staffing solution (Diao and Heching 2011); however, when we implemented these solutions in practice we often found multiple optimal (i.e., minimum-cost) solutions. In such a case, selecting a solution that causes minimum disruption to current business operations (i.e., minimum change from current staffing configurations) is desirable. Thus, in the second stage, we perform a secondary optimization in which we constrain the cost of the optimal staffing solution to equal the minimum-cost solution generated in the first stage; our objective is to minimize change from current staffing. We now provide additional details on our two-stage solution approach.

The objective of the first stage is to identify the minimum-cost staffing solution. We evaluate each proposed solution via the DESM. Evaluating alternative staffing combinations yields their associated system performance (i.e., agent utilization and request-class service-level attainment); however, exhaustive enumeration of all possible staffing combinations and evaluation of these solutions via simulation is time consuming. Further, the number of possible solutions increases as the dimensions of the problem (i.e., number of possible shifts, number of agent teams) increase. We use simulation-based optimization based upon a combination of metaheuristics—tabu and scatter search—to guide the selection of solutions for evaluation. These methods are commonly available in software packages supporting simulation-based optimization. The process for determining the optimal solution proceeds as follows. The performance of an initial proposed staffing solution is evaluated via the DESM. Its performance is measured both with respect to system cost and violation of any constraints. A next solution, selected via the metaheuristics, is selected for evaluation. Its performance is compared against the performance of previously evaluated staffing solutions. The proposed solution with best performance is marked as optimal. The procedure continues until a prespecified stopping criterion (e.g., elapsed time, elapsed number of iterations, percentage change in system performance) is reached. Glover et al. (1999) and Glover and Laguna (1977) provide details on the combination of scatter and tabu search approaches we use in our simulation-optimization methodology.

Metaheuristics guide the selection of solutions to evaluate at each step. We used the first stage of our two-stage solution approach to identify the optimal staffing solution for a large number of SDUs. Upon comparing the suggested optimal solution to the existing staffing levels, we sometimes found that the optimal solution suggested shifting agents between shifts (for example) with no change in expected performance metrics between the optimal solution and the existing staffing levels. Thus, adopting the solution suggested by the first stage would result in unnecessary disruption to business operations by modifying staffing levels with no impact on performance metrics. The second stage seeks to eliminate such situations. Its objective is to minimize disruption to existing business operations, that is, to minimize the change in staffing levels. All constraints are similar to those in the first-stage optimization; however, we add a single constraint—that the cost of the solution identified in the second stage cannot exceed the cost of the optimal solution identified in the first stage.

To date, our two-stage solution framework has been used with more than 640 global SDUs to support determining the optimal number of agents with different skills to assign to each specified shift. These SDUs continually use the framework to reevaluate staffing needs in response to changes in the delivery environment that result from factors such as changes in agent skills, customer demand (changes in arriving request volumes), or contractual agreements (changes in SLAs or changes in supported services).

Figure 5 provides one example of an SDU that improved efficiency with our modeling activities. As the

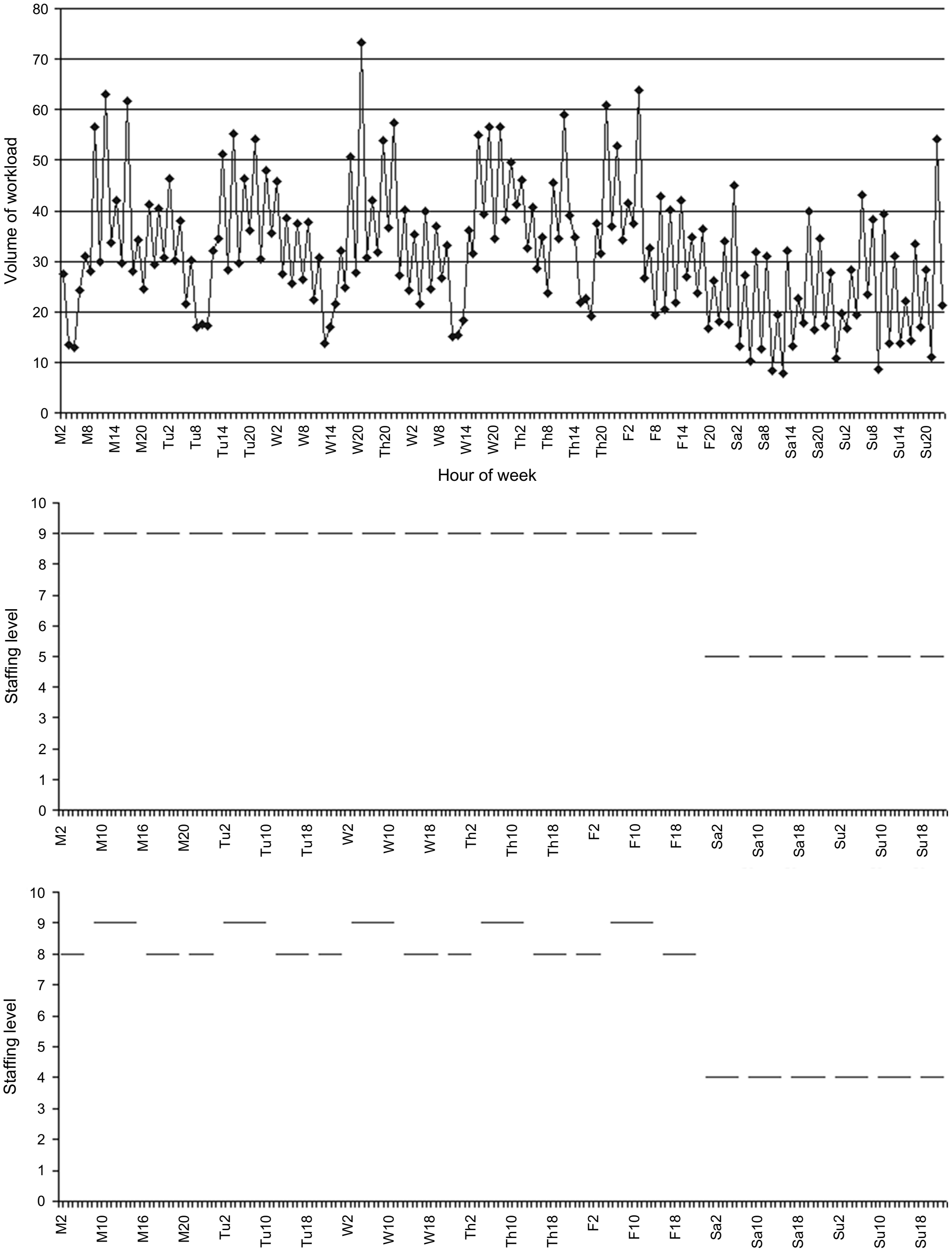

Figure 5: Our modeling activities identified opportunities for improved efficiency. This example illustrates an SDU in which staffing was fixed throughout the week and lower on the weekend. Incoming workload exhibited peaks and valleys throughout the week and dropped on the weekend. The modeling suggested a method to align shift staffing with workload with resulting lower overall cost.

top plot in Figure 5 shows, the workload peaks periodically each day and drops over the weekend. Prior to our modeling activities, nine agents worked each weekday, five agents worked on the weekend, and the team size was 36 agents. The middle plot in Figure 5 depicts this assignment of agents to shifts. Our modeling activities suggested an optimal staffing of eight agents working each weekday and four agents on the weekends, with one agent working each weekday during the peak shift only. The bottom plot in Figure 5 depicts this new staffing level. The suggested team size is 33 agents, yielding an 8.3 percent efficiency improvement.

One of the most significant technical challenges we encountered when implementing our approach is the time required to identify the optimal solution. With the multiple possible combinations of shifts and agent skills, the search space of possible solutions is extremely large and identifying the optimal solution via our two-stage solution framework required significant computational time. Often, after we had identified the optimal solution for an SDU via our framework and reviewed the recommended optimal staffing levels with management, management determined that data input changes were required, thus compounding the problem. These data changes may be caused by errors in the data initially provided (e.g., incomplete or incorrect data) or the dynamic nature of the work environment (e.g., changes in customers or services supported). Regardless of the reason, these data changes meant that all the work we did identifying the optimal solution would need to be redone.

As such, we identified three approaches to speed up identification of the optimal solution: (1) shrink the search space, (2) determine rules to quickly identify infeasible solutions, and (3) identify a good starting solution for the simulation-optimization procedure. Shrinking the search space involves eliminating a priori a set of solutions from the set of possible solutions that will be evaluated. This involves designing and introducing additional constraints that reduce the search space. Although we thought this method would be effective, we found it extremely difficult to design constraints that effectively eliminated a large number of infeasible solutions, but did not eliminate any feasible solutions. We therefore looked at other options. The second approach involved designing methods that quickly identify infeasible or suboptimal solutions as the DESM is evaluating them, and then stopping the evaluation. We introduce these methods in the form of two rules: First, we monitor the volume of backlog over the course of the system evaluation. Statistically significant growth in the backlog, measured over varying time intervals, indicates an infeasible solution. Second, we monitor the time required to complete each iteration of the DESM. One module of the DESM requires prioritizing and ordering all requests waiting to be assigned. A long queue (and thus a large number of requests that must be sorted) results in a long scenario evaluation time. Thus, time to evaluate a scenario can be used as a proxy for feasibility of a solution. The third and most successful approach that we implemented is identifying a good starting point for the two-stage solution procedure. We found that providing the procedure with a good warm start significantly reduced the overall solution time. To date, we have used heuristic methods to achieve this objective (e.g., data inspection to match skills with workload arrival patterns), but we also began exploring using approximate analytical methods as an efficient means to provide a good starting solution to our two-step solution approach; for example, see Heching and Squillante (2012).

Scaling Deployment

We needed to overcome several obstacles to deploy the service-delivery models globally, across tens of thousands of agents in hundreds of SDUs. Variability across the SDUs is a primary obstacle to scaling. Examples include how SDUs and agent teams deliver services, the deployment process steps and timing, the input data quality, and the deployment analyst skills and techniques (especially interpersonal, project management, and exploratory data analysis skills). In response, we modified our modeling approach and our approach to model deployment to address the variability that we observed across the globally distributed SDUs and between modelers and analysts. Next, we describe in depth these modifications.

Managing Deployment

After some initial implementation where deployment approaches followed broad guidelines but were strongly influenced by individual modelers, we created best-practice steps and associated timelines. Figure 6 shows the five key steps involved in deploying our models at an SDU: (1) team introduction and coordination, (2) data collection, (3) data analysis and preprocessing, (4) modeling and optimization, and (5) pilot study and key performance indicator analysis.

Our simulation project team included roles such as SDU focal point, project manager, deployment analyst, program director, development lead, senior modeler, and junior analyst. These roles jointly cover more than 90 defined tasks. For example, SDU focal points work with their managers and teams to gather three categories of data—workload, work activity, and demographic. SDU focal points provide these data to junior analysts via spreadsheets that are loaded with pull-down menus and macros, which are designed to reduce data errors. Intended date ranges for the data are explicit, and references to customers either include consistent spelling and punctuation, or are mapped (via a mapping table) to this canonical format. The result is higher data quality with less data-collection effort, and greater consistency across an increasing number of analysts.

In addition, the modeling team holds daily status calls in which the team immediately identifies and remedies any delays in deployment. Rapid escalations assure adherence to the process and timeline. This applies to the simulation team and to the SDUs under study.

Presently, we review (model) SDU composition twice a year. Initially, because we had more than 400 teams to review, we were only able to review teams annually. With the continual streamlining of our methodology, however, we are now able to review teams twice a year. As our confidence in the models grows, further improvements that allow more data reuse and reduced piloting will let us reach our ultimate goal of quarterly team-composition reviews, even as the number of teams grows and our business expands.

Input Data Quality

The workload data are drawn from ticketing systems in which arrival date and time, complexity, customer serviced, and other ticket details are recorded. We model the arrivals as nonhomogeneous Poisson processes, with arrival rates empirically determined and potentially varying by each hour of the week. Although the data are extracted from existing data sources, these data may be contaminated or difficult to interpret. When determining the arrival rates, we often must consider that the date range of the overall study contains partial ranges for some (new or discontinued) customers, or other anomalies.

The work activity data are used to compute request service times. Historically, no system records such data, necessitating data collection from each SDU. Our analysis of the data reveals that the service times follow a lognormal distribution. Gathering the work activity data and reviewing this data to ensure the quality of the data collected represent a significant proportion of the deployment effort. We learned that quickly identifying systematic errors in data collection is best; therefore, we developed a number of methods by which we could automatically analyze the data on a regular basis to scan for various data entry errors. We implemented these methods in spreadsheet tools (via macros), which was used by the focal point in each SDU either daily (at the start of data collection) or weekly (once data quality was sufficiently high) to improve the quality of the work activity data collected.

The demographic data include the team composition, the agents and their skill levels, the contractually agreed-upon SLAs, and the timing and staffing of shifts. These data are usually readily available but need to be compiled in a single place with standard formats.

We developed a Web portal where the SDU focal point would submit the data that the DESM required. The focal point would submit the workload data and demographic data only once, but would typically submit the work activity data weekly (as new data were collected). The Web portal performs a number of data quality checks, such as ensuring the consistency of customer and agent names across files and checking for data outliers. The Web portal also contains a dashboard that displays the status of the data collection, allowing the simulation team quick visibility into the status of data collection (for each of the three data types) for each SDU. Finally, the SDU focal point's submission of the files triggers a notification to the analyst to begin the data validation stage.

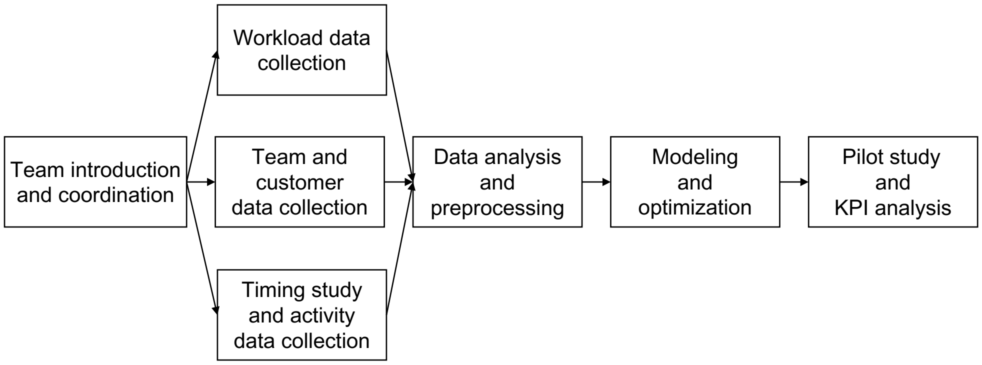
Workload data collection
Team introduction and coordination
Team and customer data collection
Data analysis and preprocessing
Modeling and optimization
Pilot study and KPI analysis
Timing study and activity data collection

Figure 6: In the five steps of the model deployment process, resources with appropriate skills are aligned to each step. We also developed training materials to yield a uniform deployment.

Visualization Techniques

Although the senior modelers established best practices early on for performing exploratory data analysis and data validation, we realized that instituting a consistent methodology would be beneficial. Further, junior analysts required greater guidance and support to ensure that data entered into the DESM were of sufficient quality.

Therefore, in the next phase, we developed automated standardized capabilities for analyzing and graphing the input data that the focal point provided. These standardized capabilities were accessible to the analyst as soon as all input data were uploaded to the Web portal. The results of the data analysis and visualization capability are provided in a combination of multitab spreadsheets and Brio reporting and graphics capabilities.

Several different analyses are provided via the graphical visualization, including box plots to visualize the distribution of the service-time data and detect the presence of outliers and simple trend charts to observe patterns in the service-request volumes. Figure 7 shows an example of a particularly useful technique. The graph superimposes shift staffing with ticket arrivals and allows an easy check for how the staffing aligns with incoming-request volumes. We did not anticipate one common benefit: the chart often reveals a mismatch in time zones, that is, the customer-specific ticketing systems often store data in a different time zone than that of the SDU. This visualization highlights the need for the focal point to provide time zone offsets, as required, to report arrivals in the team's local time.

A final contributor to data quality is requiring SDU sign off on the data submitted. Of course, SDU senior managers are not personally familiar with each customer's ticketing volume or the distribution of ticketing volumes across request classes and request priorities; however, the sign-off process requires managers and focal points to participate in reviews ensuring the validity of the data submitted, prior to the start of the extensive data analysis and time expended in executing the DESM. This step required strong executive support. Combined with the analytical techniques described previously, this managerial data sign-off step reduced the rework as compared with the rework required in the early phases of DESM deployment.

Model Consolidation

In the next phase, to further mitigate the impact of the diverse backgrounds of the analysts, we introduced additional features that made the simulation and optimization tool simpler and more robust. Multiple senior modelers developed our models over several years. Consequently, the models differed. Some of these differences were superficial (e.g., warm-up time in minutes in one model and days in another); others were more significant (e.g., a range of prioritization rules from which the modeler can select). To address the differences, we unified the look and feel of the user interfaces and model logic. We also split the input parameter screen into two screens to simplify the number of choices required by the modelers.

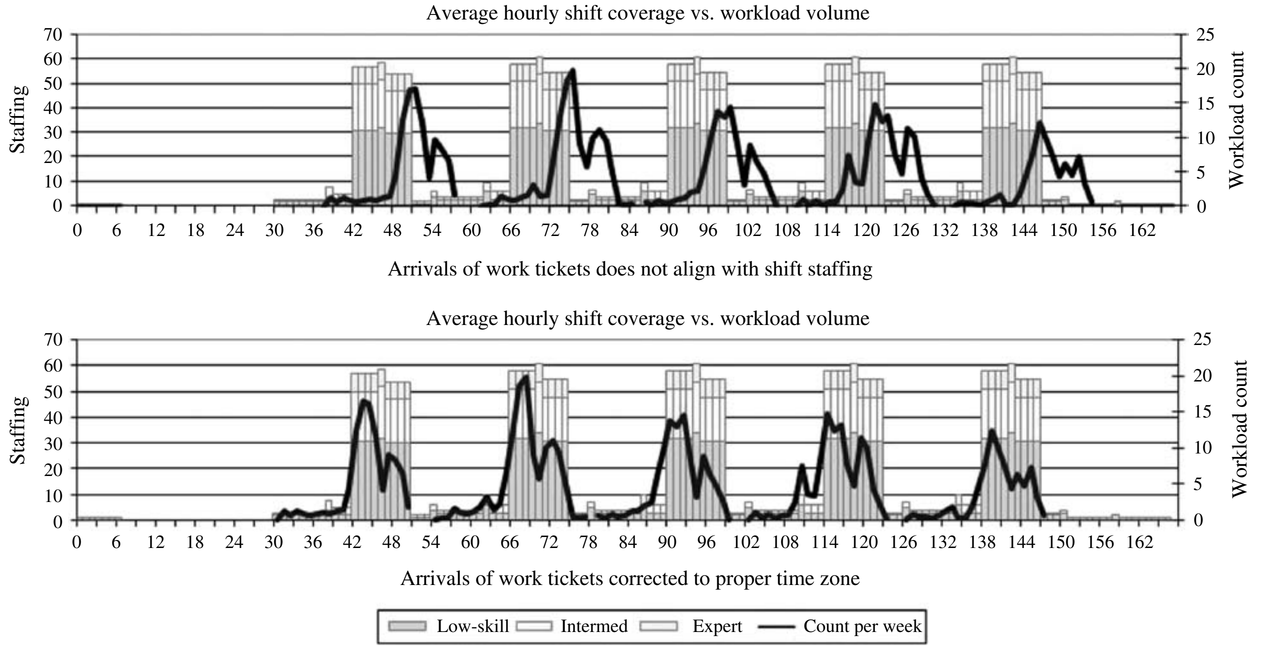

Figure 7: We use visualization techniques to improve the quality of the data collected. In this figure, we depict agent shift schedules plotted with workload volume. Misalignment of these two data sets, shifted by a fixed number of hours, highlight that the timestamps in these data sets do not reflect the same time zone.

The first screen contains the parameters that the modeler must specify (e.g., number of shifts, team working hours, types of request classes the SDU supports). The second screen contains more advanced options (e.g., warm-up period, stopping criterion), which we set to default values but allow the modeler to modify. This process resulted in model consolidation, resulting in three core models and allowing us to deploy 11 types of SDUs.

Deployment Team

Given the significant effort we invested in standardization and automation, one may conclude that the modeling and staffing optimization process is essentially a turnkey deployment. The DESM takes as input a data file output by the preprocessor engine (embedded in the data portal) and generates a complete, formatted final report that contains details of the study and conclusions. The variability we encounter (particularly, input data sources, data availability, and interpretation of the data and simulation results), however, requires ongoing attention from a skilled analytics professional.

To more firmly embed this decision support within operations, we expanded the analyst team to those located at the global service-delivery centers. Using local resources helps centers' management teams view the analytics as an integral part of center management rather than as a one-time initiative. Using local talent also provides required language skills and cultural sensitivity and alerts the modeling team to more subtle differences across delivery centers, such as national holidays and legal policies.

To identify appropriate local analyst resources, we looked for respected employees with service-delivery experience and an interest in analytics. Their backgrounds have ranged from individuals with a bachelor's degree in industrial engineering to individuals who have taken only a single (long-forgotten) statistics class. To bring them to the same level of understanding of the process and models, we developed and conducted intensive hands-on training in three areas: (1) basic knowledge of statistics, simulation, queuing theory, and optimization; (2) operating instructions, including deployment steps and how to use developed data templates, visualization, and models; and

(3) applied exercises that use real data and examples to reinforce the concepts and techniques.

We conducted the training using both classroom and virtual training. Two weeks of classroom instruction are followed by a third week of applied exercises that are conducted remotely. The focus of the training is on the first and most stable model, which covers the majority of the teams. Approximately eight months later, we conduct follow-on sessions to train the modelers on the details of the other models.

In addition to the training sessions, close mentoring was provided to assure thorough and consistent analysis. Each new analyst was assigned a senior modeler mentor to assist during the first launch (during which the new analyst applied the model to several SDUs). Style and expectation differences led to uneven support; therefore, we established a weekly senior mentor call at which we discussed best practices for mentoring the new analysts, and randomly selected from each others' protégé's analyses and discussed the analysis approach. The atmosphere of collegial challenge led to stronger and more uniform mentoring and additional learning across this experienced team.

The increasing size of the deployment team has resulted in an increase in the number of deployments (more than 250 in 2012). We measured productivity of the overall work effort as deployment per person, including nonanalyst support staff who performed tasks such as code maintenance or project management. Because of the investment in building capability, this metric has been fairly static.

With all global analysts fully trained and experienced and the deployment process further streamlined, we expect deployments per person to double. For example, we have reviewed how often piloting a recommended solution resulted in a change to that recommendation. We now determine whether to do a pilot based on the recommended staffing change; for small staffing changes, we implement the change but omit the pilot. This saves five weeks of deployment time, thereby increasing analyst capacity by more than 30 percent in these cases.

Business Results

The impact to our business has been significant in several ways. First and foremost, our SDUs are now consistently staffed correctly to deliver the service levels to which they are committed. As our modeling work has consistently demonstrated, approximately two-thirds of our teams were not correctly staffed as compared with the staffing levels suggested by the models; this incorrect staffing is equally split between over- and understaffed SDUs. Prior to our modeling effort, staffing for these SDUs was primarily performed based on ratios of staff to supported components, such as servers, storage, and consoles. These ratios were typically averages derived from prior experience; however, workload was seldom explicitly considered. Achieving the proper staffing levels for our SDUs has greatly reduced wasted resources in overstaffed SDUs and reduced the risk of under delivery (and hence penalty payments) in understaffed SDUs. Although the number of understaffed SDUs was consistently equal to the number of overstaffed SDUs, the percentage of overstaffing in an SDU tended to be higher than the percentage of understaffing across the environment. We believe this is because SDUs that are significantly understaffed tend to struggle operationally; therefore, they attract management attention, which leads to giving them additional resources over time. Overstaffed SDUs are not similarly visible in day-to-day operations.

As we previously mentioned, understaffed SDUs tend to miss SLAs. Most SLAs in this environment set guidelines for the percentage of tasks that must be completed in a fixed interval (e.g., 95 percent of priority 3 incident tickets resolved within three business days). Because this work is fundamentally a queuing system, an SDU with insufficient resources will miss the targets some fraction of the time. Even SDUs with the correct number of resources will struggle to make targets if their resources are not deployed at the right times, especially on tasks with relatively shorter SLA targets. Therefore, because our models determined the correct staffing levels and shift placements, we achieved better performance against SLA targets. This was also noticeable on SDUs that had the correct staffing numbers, but had the resources deployed suboptimally—a fairly frequent occurrence. These SDUs also benefitted from improved shift placements. For example, many SDUs struggle with priority 2 incidents because they tend to have a resolution target of 8–12 hours; yet, they do not generate the intense focus of priority 1 incidents. Optimizing shift patterns—even when the staffing level is correct—can reduce the probability of tickets arriving when agents are not available to respond within the required time.

When quantified, these benefits amounted to approximately $52 million in documented cost savings and avoidance across the 640 SDUs and 13,000 agents that we modeled. Savings were achieved through net reductions of overstaffed situations; costs avoided were based on reductions from historical levels of penalties that had been paid to customers for breaches of contractual service levels. Cost savings from overstaffed situations typically ranged between three and 12 percent of labor costs, with occasional SDUs reaching savings as high as 15–18 percent. In understaffed situations, additional labor costs typically ran from three to seven percent, with many of these costs offset by the avoidance of penalty payments. Penalty payments can vary widely; a single monthly penalty can range from $5,000 to $250,000 (or more in rare cases); $52 million is a net calculation across both staffing and penalty avoidance. From the beginning of the project, our corporate finance teams meticulously tracked these benefits, which we consider important internal measures of the project's success.

In addition to the monetary benefits, the modeling work has provided important operational benefits. Using the models has educated our SDUs on the use of data-driven analytics for fact-based decision making. Although our SDUs have always used measurements to drive decision making, our models have brought them a new level of sophistication. This is providing long-term, sustainable benefits to the business, and prompting SDU management to ask deeper questions, thereby driving the expansion of analytics in the business over time.

Lessons Learned and Future Work

We previously covered many of the lessons learned; therefore, we will highlight only one major lesson here. During the first two years of the project, we learned that to enable massive deployment on the scale that we have demonstrated, developing processes and automation for end-to-end support of the analytical methods was necessary. That is, although the analytical methods were necessary, they were not sufficient to enable us to drive repeated use of the models across the globe in a standard and consistent manner. The ultimate success of this program depended on a support structure that was based on rigorous process and project discipline.

Going forward, we aim to expand the use of these models in three critical aspects of our strategic outsourcing business:

- Customer engagements: The models can answer many detailed questions that our current tools cannot, especially with respect to the impact of SLAs on resource requirements.
- Business case generation: The models can help us accurately understand the real impact of changes in our environments, and avoid aggregation errors that lead to overestimating benefits.
- Continual improvement: The models are ideal laboratories for sensitivity analysis and bottleneck detection, and for testing new ideas.

Conclusions

Service-delivery modeling and optimization is a complex problem that involves multiple service levels, skill sets, request classes, and service times. In this paper, we describe a comprehensive and scalable end-to-end analytical methodology that we developed and implemented in a large global IT service-delivery environment. This analytical methodology provides predictive insight and prescriptive solutions to the problem of staffing service-delivery groups in this complex environment. Our solution has been deployed globally to more than 640 service-delivery teams, and has yielded more than $52 million in cost savings and avoidance to date. In addition, our analytical methodology is being expanded beyond delivery to our sales, investment, and quality areas.

Acknowledgments

This paper was a finalist in the Innovative Applications in Analytics Award competition in 2012.

References

Aksin Z, Armony M, Mehrotra V (2007) The modern call center: A multi-disciplinary perspective on operations management research. *Production Oper. Management* 16(6):665–688.

Anerousis N, Diao Y, Heching A (2010) Elements of system design optimization in service quality management. Kiriha Y, Granville LZ, Medhi D, Tonouchi T, Kim M-S, eds. *Network Oper. Management Sympos.* (Institute of Electrical and Electronics Engineers, Washington, DC), 48–55.

Atlason J, Epelman MA, Henderson SG (2008) Optimizing call center staffing using simulation and analytic center cutting-plane methods. *Management Sci.* 54(2):295–309.

Bouzada MAC (2009) Scenario analysis within a call center using simulation. *J. Oper. Supply Chain Management* 2(1):89–103.

Brown L, Gans N, Mandelbaum A, Sakov A, Shen H, Zeltyn S, Zhao L (2005) Statistical analysis of a telephone call center: A queueing-science perspective. *J. Amer. Statist. Assoc.* 100(March):36–50.

Cezik MT, L'Ecuyer P (2008) Staffing multiskill call centers via linear programming and simulation. *Management Sci.* 54(2): 310–323.

Diao Y, Heching A (2011) Staffing optimization in complex service delivery systems. Accessed October 1, 2014, http://www.simpleweb.org/ifip/Conferences/CNSM/2011/papers/85739_1.pdf.

Diao Y, Heching A, Northcutt D, Stark G (2011) Modeling a complex global service delivery system. Accessed October 1, 2014, http://www.informs-sim.org/wsc11papers/062.pdf.

Feldman Z, Mandelbaum A (2010) Using simulation based stochastic approximation to optimize staffing of systems with skills based routing. Johansson B, Jain S, Montoya-Torres J, Hugan J, Yucesan E, eds. *Winter Simulation Conf.* (Institute of Electrical and Electronics Engineers, Washington, DC), 3307–3317.

Gans N, Koole G, Mandelbaum A (2003) Telephone call centers: Tutorial, review, and research prospects. *Manufacturing Service Oper. Management* 5(2):79–141.

Glover F, Laguna M (1977) Heuristics for integer programming using surrogate constraints. *Decision Sci.* 8(1):156–166.

Glover F, Kelly J, Laguna M (1999) New advances for wedding optimization and simulation. Farrington PA, Nembhard HB, Sturrock DT, Evans GW, eds. *Proc.* 1999 *Winter Simulation Conf.* (The Society for Computer Simulation International, San Diego), 255–260.

Heching A, Squillante M (2012) Stochastic decision making in information technology services delivery. Faulin J, Juan AA, Grasman S, Fry MJ, eds. *Decision Making in Service Industries: A Practical Approach* (CRC Press, Boca Raton, FL), 3–36.

Robbins TR, Harrison TP (2008) A simulation based scheduling model for call centers with uncertain arrival rates. Mason SJ, Hill RR, Moench L, Rose O, Jefferson T, Fowler JW, eds. *Winter Simulation Conf.* (Institute of Electrical and Electronics Engineers, Washington, DC), 2884–2890.

Yixin Diao is a research staff member at the IBM Thomas J. Watson Research Center. He received his PhD in electrical engineering from Ohio State University in 2000. He has published more than 80 papers in systems and services management and is the co-author of the book *Feedback Control of Computing Systems* (Wiley 2004). He is the recipient of the 2002 Best Paper Award at IEEE/IFIP Network Operations and Management Symposium, 2002–2005 Theory Paper Prize from the International Federation of Automatic Control, 2008 Best Paper Award at IEEE International Conference on Services Computing, and 2014 Best Paper Award at IEEE/IFIP Network Operations and Management Symposium.

Aliza Heching is a research staff member in the Mathematical Science Department at the IBM Thomas J. Watson Research Center. She received her B.A. in mathematics from City University and her M.S. and Ph.D. degrees in operations research from Columbia University. She joined IBM Research in 1998. Her research interests include measuring and optimizing the design, performance, and operational efficiency of service systems. She has consulted extensively with both internal IBM divisions and external clients.

David Northcutt is a retired IBM distinguished engineer with more than 30 years of industry experience in the areas of applied statistics, data presentation, modeling, estimation, and continual improvement techniques. He holds an M.A. in economics from Northwestern, an M.S. in computer science from University of Illinois at Chicago, and an M.S. in statistics from Rutgers University. He is a part-time consultant working with clients worldwide. He is an INFORMS Certified Analytics Professional (CAP), an American Society for Quality (ASQ) Certified Quality Engineer, and a senior member of ASQ.

Rodney Wallace is an IBM program manager responsible for directing teams of senior technology experts and IBM researchers in developing and deploying innovative business analytics within IBM Global Technology Services division to optimize and improve the quality of the services delivered to IBM clients. He holds a B.S. in mathematics from University of Georgia, an M.S. in mathematics from Southern University, and a D.Sc. in operations research from The George Washington University.