Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

MODELING A COMPLEX GLOBAL SERVICE DELIVERY SYSTEM

Yixin Diao
Aliza Heching

IBM T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598, USA

David Northcutt
George Stark

IBM Global Technology Services
294 Route 100
Somers, NY 10589, USA

ABSTRACT

Enterprises and IT service providers are increasingly challenged with improving the quality of service while reducing the cost of service delivery. Effectively balancing dynamic customer workload, strict service level constraints, and diverse service personnel skills challenges the most experienced management teams. In this paper we describe a modeling framework for analyzing complex service delivery systems. The interaction among various key factors are included in the model to allow decision-making around staffing skill levels, scheduling, and service level constraints in system design. We demonstrate the applicability of the proposed approach in a large IT services delivery environment.

1 INTRODUCTION

In recent years, the IT services industry has faced continual pressure to improve the quality of its services, while doing so at reduced cost to its customers. To measure quality of services delivered, the IT services delivery industry is converging toward a set of commonly used metrics including, but not limited to, equipment availability, time to resolve incidents, etc. In an effort to improve the quality of their services, service providers need to adopt a consistent and continual focus on their internal processes, the skills of their people, the organizational structure, and so on. On the other hand, the IT service industry has been traditionally labor intensive. Although a wide range of management tools is available to support even the most complex management tasks, the overall service management process still relies on humans to make important decisions, perform complex administration tasks, and monitor the performance and effectiveness of the overall process. This paper describes a modeling framework to support decision making in a complex service delivery system including the cost of a service (as manifested by the staffing level) and the corresponding quality metrics (subject to the service level agreements). In particular, we model the service delivery organization through the use of a discrete event simulation model that considers the various factors typically encountered in a service provider environment.

We consider a service delivery system in the context of global IT services delivery. Global delivery refers to a model for delivering IT services where the services provider may provide services from either on-shore or off-shore locations to customers who are globally located. The managed systems represent customer IT systems that are managed by the services provider. These systems (e.g., servers, networks, application, business processes) may be owned by the customers and located on customer sites; alternatively, they may be owned and located on provider sites on behalf of customers. The managing or management systems represent the service delivery centers that receive service requests from the customers and interact with the managed systems regarding the requests. The service delivery centers operate based on a cohesive set of management processes, such as Service Desk which serves as a contact between service providers and service users, Incident Management to quickly restore normal service operations in the event of failure, Change Management to manage changes to the infrastructure and the services, and Service Level Management to ensure Service Level Agreements (SLAs) are met with minimum impact on service quality

978-1-4577-2109-0/11/$26.00 ©2011 IEEE

(Office of Government Commerce 2007). Remote skills represent the service personnel located in global delivery centers who respond to customer requests, but do not directly interact with the end customer. It is important that the global model of delivery is transparent to the customers; the service delivery centers are the central point of contact and management for the end customers.

Such a distributed network of delivery offers advantages to customers including (i) allowing the customers to take advantage of qualified local skills in each of the respective local locations, (ii) providing round the clock coverage and services by different support teams in each of the different locations around the world, and (iii) improving the resiliency due to the distributed nature of the support teams and data centers. On the other hand, the global delivery model imposes a challenge for service providers: how to improve the quality of its services in a most cost effective way. While lowering the delivery cost may be achieved by servicing multiple customers from the same delivery center, a major challenge faced by the service delivery provider is to determine the interaction between the arrival rates of the different requests for service, the team specific service rates for the different types of service requests, the available skills from the service agents, and the different service level target requirements, in order to determine the required staffing levels and shift schedules that minimizes the overall cost of delivery.

The problem that we describe falls in 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, Koole, and Mandelbaum 2003) and (Aksin, Armony, and Mehrotra 2007) provide detailed surveys of the analytical approaches that have been undertaken. The most common approaches are to either simplify the topology of the network or simplify the routing schemes. However, none of them are desirable in a service delivery environment where both the network and the routing schemes are complex and the service providers are seeking practical solutions but 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 the 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, Epelman, and Henderson 2008) considers a multi-period 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 LaEcuyer 2008) extends this approach by applying it to large problem instances and developing heuristic methods to handle the numerical challenges that arise. (Feldman and Mandelbaum 2010) uses 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 enter as costs in the objective function, and use simulation to evaluate service level attainment. (Robbins and Harrison 2008) considers 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 the 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 change in handling time distribution, change in call volume, or changes in SLA constraints. (Anerousis, Diao, and Heching 2010) uses simulation to study how optimal staffing in a service delivery organization is affected by different operational scenarios that reflect a diverse skill base, the presence of service level objectives, and incoming work with varying levels of complexity.

In this paper we propose a simulation-based approach to determine minimum staffing requirements while meeting contractual service quality commitments in a global service delivery system. It is worthy to note that the focus of this paper is not on proposing a new simulation or routing algorithm, but on how the standard approach can be applied (and deployed in a large scale) to a global service delivery environment. By analyzing system dynamics and business requirements, we identify the key service system factors that have the first order effects on the modeling objectives. The proposed approach requires a reasonable set of operational and demographic data which lends its applicability in real service delivery environments. Meanwhile, following the methodology in (Glover 1977), (Glover and Laguna 1977), and (Glover, Kelly, and Laguna 1999), we use scatter search combined with tabu search for staffing level optimization. The model is implemented and deployed in a large services delivery provider with worldwide delivery locations and international customers.

The remainder of this paper is organized as follows. Section 2 discusses the background for service delivery systems and the challenges for modeling and decision making. Section 3 presents the architecture and components of the proposed service delivery model. Section 4 describes the model implementation and deployment results. Our conclusions are contained in Section 5.

2 GLOBAL SERVICE DELIVERY

In Global Service Delivery, customers contract with the service provider on a menu of IT services such as security patch management and data restore management. This contract specifies the scope of services (e.g., number of servers, number of users), the locations from which services will be provided (onsite, offsite), and the measure of service quality (i.e., service level targets). The service delivery provider responds by assigning each contracted service to a delivery location and maintains a team of service agents to respond to customers' service requests. The agents typically are differentiated with respect to depth and breadth of skill, where breadth of skill refers to the range of IT areas in which the agent has knowledge and depth of skills refers to the level of knowledge mastered by the agent in each of these IT areas. Agents are grouped into teams where all agents in a team have common breadth and depth of skills.

We now describe the workload management process after customers contract for service and once customer requests begin to arrive to the service delivery provider. We describe the process within the scope of a Service Functional Unit. A service delivery center typically supports multiple service functions such as platform support, storage management, and console monitoring. One functional unit represents a unit within one service function area. Customers interact with the service functional unit through the service request management system, which coordinates service request creation, queueing, assignment, and closure. Based on the nature of the required services, service requests are assigned to different service delivery units which are comprised of a team of service agents providing a set of services to the customers.

The teams of service agents responding to customer service requests may be colocated or virtual. (In a virtual team, all agents are not physically colocated but they function as a team and their performance is jointly measured.) As customer requests arrive to the provider, the requests are routed to the appropriate delivery location and service delivery units based upon a number of attributes of the request including: customer, problem description, required skills, and time of arrival. Time of arrival is often an important attribute due to the nature of global delivery and the ability of the provider to "follow the sun," i.e., to utilize agents in each geographic region during their normal business hours. Arriving requests are prioritized and then assigned to an agent in the team.

In addition to the scope of service requests that the provider will service for the customer, customer contracts specify service levels associated with each of these service requests. Service levels are a measure of quality of service delivery. Although many types of service level agreements exist, the most common service level agreements take on the following form. They specify the following main terms regarding response to a service request (i) scope of agreement (ii) target time (iii) percentage attainment and (iv) time frame over which service will be measured. For example, a service level agreement may state that 95% (percentage attainment) of all severity 1 tickets (scope) that are opened over each one month period (time frame) must be resolved within 3 hours (target time). One will typically find a large number of service level agreements associated with each customer contract.

Customer service requests can be broadly classified into two types: primary requests and project requests. Primary requests are characterized by relatively short service time (typically, minutes or hours) and short target time (typically, hours or days), and in most cases require a single agent to complete the request. Examples of primary requests include problem tickets, change requests, and maintenance work. Project requests are characterized by requests that are composed of a sequence of tasks and that may require the coordination of a number of different delivery units where different units are responsible for different tasks in the overall project request. There may be dependency relationships between the different tasks. Tasks within a project often take weeks or months to complete. In many cases, the agents who service the project workload are different from those who service the primary workload. This is due to the different skills required. In other cases, these teams of agents are separate due to the differences in cadence and arrival processes for these two types of workload and the ease in management that is introduced by separating these two types of workload.

In this paper, we focus our study on service delivery units that respond to the customer primary requests. Customer requests arriving to the system have a number of attributes including the associated customer, priority (severity), and required skills. The combination of customer, priority, and request type determine the target response time and associated percentage attainment. With this information, all arriving service requests are assigned to a priority class and we assume that the arrival rates of workload to the different priority classes are independent. The purpose of modeling the delivery system is to support model-based decision making and what-if analysis. For example, we can use the model to help determine the minimal number of agents required in each service delivery unit to meet the service level requirements from all serviced customers.

There are a number of complexities in this system. First, the number of classes of requests arriving to the system is large, since the requests are differentiated by the different attributes, and the request arrival rates are non-stationary and vary over the hours of the day and days of the week. Second, the number of service delivery units is large since agents with different skill sets are assigned to different service delivery units, and agents are differentiated both by breadth and depth of skills. Furthermore, different agents are working on different shift hours. Third, the agents will take random breaks at random times throughout the day. Although the total duration of these breaks is assumed to be known (e.g., an agent is assumed to be available 85% of his total shift and the remaining 15% of the shift he is in breaks / meal break) the exact timing of these breaks is not known. Finally, the service level attainment level is measured against system performance over extended time periods, typically one month, rather than against performance of each customer request. In addition, the service target time can be either in the unit of calendar hours or business hours; in the latter case, a business calendar is required. It is due to these modeling complexities as well as the inherent stochastic nature of the problem that we choose a modeling framework based upon discrete event simulation.

3 SIMULATION MODEL

In this section we describe the discrete event simulation model in detail. Figure 1 illustrates the flow and major complements of the model, which includes service requests, service delivery units, dispatching engine, and performance calculation.

3.1 Service Requests

We use service requests to model the incoming customer workload that the service delivery unit needs to serve. We characterize the service requests through the following attributes: arrival time, customer, work type, tooling, severity, complexity, and service time.

Arrival time defines the time when the service request is generated. The arrival time defines the start of the service and initiates the calculation of whether the service level target is met. *Customer* is an attribute that is used to differentiates the requests since different customers typically have different service

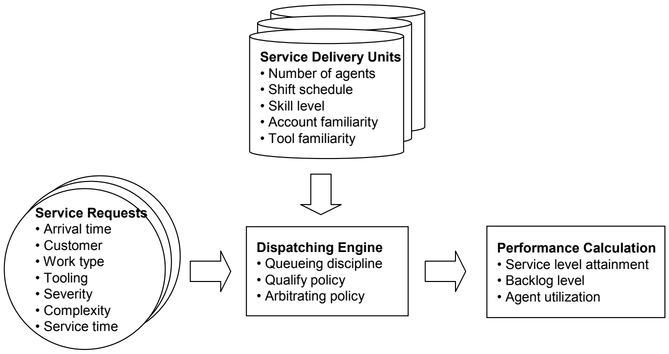

Figure 1: Architecture of the service delivery system model.

level requirements. Note that in a global service delivery system the service delivery units can handle service requests from multiple customers. This helps to better leverage the agent skills and improve their utilizations.

We define *work type* based on the nature of the work (e.g., problem, change, maintenance) but not the source of where they are from. Indeed, the service requests may be generated from different sources. Some requests are generated as a result of telephone calls to call centers, or created directly by customers via email or online ticketing systems. Other requests may be automatically generated by monitoring systems; that is, systems are programmed to trigger requests in the case that the warning thresholds are violated. In addition, the service delivery units may have scheduled work that needs to be performed on a regular basis, which represents another form of workload. However, we model each class of work based on the type of the work but not the source, because different work types have different service process, different service time, and different service level agreements.

We use *tooling* to refer to the different technical tools or platforms that a service agent needs to grasps in order to be able to work on a service request. For example, in UNIX platform support, different skills are required for AIX versus HP-UNIX or Solaris. *Severity* classifies the service requests based on their urgency or priority. Although there is actually a difference between urgency and priority, we use the term severity to correspond to the service level agreements where a different penalty is given for different violations. *Complexity* classifies the service requests based on the complexity of the work and implies the depth of the skills. The introduction of complexity groups the service requests into several buckets so that the *service time*, the time required to handle the service request, can be quantified with smaller variance.

Upon service request characterization, we derive the workload arrival patterns using the historical data maintained by each service delivery unit. Typically, we use six months of the most recent data unless any drastic change has occurred - in that case, we use the most recent stable period of data. Note that the service delivery environment is highly dynamic with supported customers changing often or, for long term customers, the base of services provided to the customer changes over time. For example, additional services may be added or the number of systems supported may increase. As such, in cases where longer periods of historical data are available, we still prefer to use the most recent data that reflect the current service operation. We also derive the workload arrival patterns by customers and by work types. This is because different customers and work types have different arrival patterns. In addition, having separate arrival patterns also makes it easy to add or remove customers or work types from the service functional unit, when changes in customer contracts or service organization occur.

We characterize workload arrival using weekly patterns to capture the workload variation over the hours of the day and the days of the week. For example, typically, emergency system repair work would be performed during the week but all planned maintenance work would be scheduled for the weekends. However, the pattern would be different for different work types and different customers. The variability in the arriving workload is stochastic in nature over short periods of time (i.e., hours), but exhibits a trend or repeatable pattern over longer periods of time (e.g., days, weeks, or months). We decide to capture the arrival pattern on a weekly basis. That is, the arrival rate distribution is estimated for each of the 168 hours of the week, where the arrival rate is assumed to follow a stationary Poisson arrival process within each of these one hour time periods.

The reasons that we are not taking the arrival patterns over a longer horizon are as follows. First, due to the dynamic nature of the services delivery environment, it was typically difficult to obtain long periods of historical data that are stable enough to derive the seasonal patterns. Second, seasonal workload changes do exist, and so does the flexibility of certain non-demanding workload (e.g., documentation update, knowledge transfer) that can be held during the peak season. Third, scheduled overtime can be used to meet certain excess demand during periods of high demand, though overtime is not relied upon too extensively due to either regulatory constraints or the already long shifts that the agents have. Finally, the agents' vacations can be mostly scheduled during the time periods that were known for lower volumes of service requests, coincident with the time when most of the customers are on vacation. For example, for a group of service delivery units in Europe we observe average rate of vacation of 10% each month of the year with the exceptions of August and December where average rate of vacation across all agents is observed to be approximately 17%. This is aligned with the decrease in service requests for these two months; in August customers typically schedule vacation that results in a lower volume of service requests while December is a traditional month for so-called "change freezes" (customers freeze any changes to their systems other than critical changes), again resulting in reduced volume of service requests. In summary, although not as technically advanced as it could be, we feel that having a weekly workload pattern is practically sufficient to model the global service delivery.

Besides deriving the weekly workload arrival patterns, we also use the historical service data to derive the percentage of tickets for which different tooling is required, the percentage of tickets with different severity levels, and the percentage of tickets with different complexity. We assume all of these percentages are independent.

In contrast to the service request arrival data, we found there were no available data sources that recorded the handling times for the service requests processed by the service delivery units. Since service quality is typically measured by the total time to resolve the service requests, the service delivery units maintained accurate records regarding the end-to-end time each service request spent in the system (i.e., the time from creation of a service request until the service request is resolved). However, the simulation model requires an estimate of the true effort where a request is being serviced. To gather this information, one month long timing studies were performed in each of the service delivery units, where a time recording application was installed on the desktop of each of the agents. The tool was used to record the start, stop, and pause time for each activity performed by the agents. After collecting the data, we model service time distribution by work type and by complexity, but not by customers since we find the service times are similar across different customers from the same delivery unit. Similar to the findings of (Brown, Gans, Mandelbaum, Sakov, Shen, Zeltyn, and Zhao 2005) and supported by the theoretical work of (Ulrich and Miller 1993), we find that the distribution of the service times is well modeled by a lognormal distribution.

3.2 Service Delivery Units

Service delivery units contain the service agents and are modeled from the following aspects: number of agents, shift schedule, skill level, account familiarity, and tool familiarity.

Number of agents indicates the size of the service delivery unit and the *shift schedule* specifies the operation hours of the delivery unit. Depending on the customer contract, the service functional unit may only need to have one shift to provide customer support in normal business hours, or it may have multiple shifts to provide 24x7 support. In the latter case, the shift schedule may also be different at different weeks. However, we specify all agents within the same service delivery unit to have the same shift schedule.

The service agents within the same delivery unit are organized with the same breadth and depth of the skills. This is indicated by the identical *skill level* for all the agents within the same unit. Large IT

Service Delivery organizations have a very wide distribution of skill levels. If one were to randomly assign a service request, the result would exhibit considerable variance, both in the quality of the work and the length of time to complete. Having skill based delivery units applies uniformity in the service delivery process: simple work that arrives to the system is sent to the basic delivery unit, while difficult work is sent to the expert unit. Besides skill level, the delivery unit can also be organized based on *account familiarity* (i.e., the capabilities to service the same set of customers, which can be one or multiple) or *tool familiarity*. All of the above data elements can be easily obtained from the so-called "demographic data" of the service delivery unit.

3.3 Dispatching Engine

The dispatching engine assigns the service requests to the agents in the service delivery unit following a set of rules. The *queueing discipline* specifies the order of which the arrived service request will be processed by the service agent. A simple policy could be priority based queueing; the service request with the highest severity level gets serviced first. However, since the service level targets can be quite different for different customers for the same severity level, an earlier deadline first policy is preferable.

The *qualifying policy* determines the set of service delivery units that are qualified to handle the arrived service request. This is based on the required skills (i.e., the skill level, account familiarity, and tool familiarity) and the shift schedule (i.e., the current simulation time must be during the active shift time of the delivery unit). We note that it may be the case that all agents in a service delivery unit are occupied with tasks that were assigned earlier. However, if the request that is currently at the head of the queue is of higher priority than a request currently being handled by an agent, the request currently being handled can be preempted and returned to the central queue.

Next, we define the *arbitrating policy* that selects the service agent among the qualified units. The objective is to balance the request assignment to each of the qualified units. Although a simple round robin rule can be used, it does not always achieve a balanced workload assignment. The reason is that certain units are unique in certain ways (either with special skill or special shift schedule) and should be "reserved" instead of "abused." To balance the utilization, we use the following algorithm:

1. Define a $0-1$ "skills matrix" where each row represents one service delivery unit and each column represents each required skill. A one in position (i,j) indicates that service delivery unit i is able to respond to the request with skill j requirement; otherwise, the cell's value will be 0.
2. For each service delivery unit, calculate its uniqueness score as follows. For each row i in the skills matrix, compute $Q_i = \sum_{j=1}^{J}(i,j)/\sum_{i=1}^{N}(i,j)$. This value provides a "uniqueness score" for service delivery unit i.
3. Compute the overall utilization of the service delivery unit, U_i, where the utilization is measured as the total time that the different agents in the service delivery unit have been occupied for the given simulation trial divided by the time that has expired since the start of the simulation trial. This value is recalculated each time a service request need to be assigned.
4. Calculated uniqueness weighted score as $Q_i \cdot U_i$. An agent from the qualified service delivery unit with the lowest uniqueness score is assigned the request at the head of the queue.

3.4 Performance Calculation

The main performance indicator of the service delivery model is the service level attainment level, an indication of whether the service level agreement can be met given the current service workload and service delivery unit organization. At the end of the simulation trial, each service request is examined to determine whether it has met the service level target or not. A business calendar will be used if the target is defined in terms of business hours. Afterward, the attainment level is calculated by customer and by severity level.

The second performance indicator is the backlog level. A service delivery model may have a long backlog if the service requests are not getting serviced during the simulation trial. This could occur if the staffing levels are too low and low priority service requests never make it to the front of the queue or, if they do make it to the front of the queue, they are preempted by higher priority service requests. Since service level attainment levels are only calculated against completed service requests, we rely on the backlog level metric to indicate the internal model health, which should only be a very small fraction of the total number of requests created during the simulation trial.

Finally, we also measure the agent utilization and service delivery unit utilization to examine if the workload has been dispatched equally among the units and if the agents are over- or under-utilized given the service workload.

4 MODEL IMPLEMENTATION AND DEPLOYMENT

We implemented and deployed our simulation model at a large service delivery organization. Our deployment covered approximately 300 service functional units including 1,000 service delivery units and 8,000 service agents in 6 service function areas. These service functional units are located in ten geographic regions, with multiple service delivery centers in each region.

Our model operates at the service functional unit level, where one discrete event simulation model is used for one functional unit. Although a service request may be routed among multiple functional units, for a well organized service delivery organization, the percentage of mis-routed requests are extremely small. Therefore, we can assume no dependencies between the workload or the agents in different service functional units, and model each of the functional units independently. In this section we describe in detail how we gathered data, built the model, and used the simulation-optimization framework to determine sufficient staffing levels that would meet customer service quality and geographic specific utilization constraints.

4.1 Data Collection

We collect three types of data: the workload data (the arrival of different types of service requests), the effort data (the handling time for the different service requests), and the demographic data (the service quality commitments, the service agent shift schedule, etc.)

We start from collecting the demographic data. We gather information regarding the number of agents and the level of skills in each functional units as well as the existing shifts that are currently in place. Another set of the demographic data is the service level targets for each customer and at each severity level. The shift schedules often vary significantly in size and configuration across the different functional units. The total number of hours that an agent can work per day and per week are also subject to local regulations.

We conduct four-week timing studies at each service functional unit to gather the effort data. During this period, agents are required to log every work related activity that they perform in a time capture tool specially designed for this purpose. The tool is designed so that the agent can click a button to start an activity, to pause it (e.g., waiting for additional information from the customer), to resume it, or to close it when the request is finally resolved. At the end of timing study we conduct data cleaning to remove the outliers with unusually long duration (e.g., the agent forgot to pause the activities at the end of the working day). We also handle multi-tasking (i.e., the agent is working on several activities at the same time) by dividing the overlapped time among the concurrent activities. This is because we are modeling the true effort time. The primary use of the effort timing data is to derive the service time distribution. However, we also find not all historical data are available for all work types; in this case the timing data are also used to derive the workload distribution.

Finally, we collect the workload data to derive the workload arrival patterns. Typically, the historical workload data are saved in customers' ticketing systems (such as Maximo, ManageNow) located at different locations with different time zones. Prior to creating the arrival distributions, we convert all time stamps to the local time zone of the functional unit. This is to align with other time information used in the model (e.g., shift schedules, business hours for calculating the service level targets).

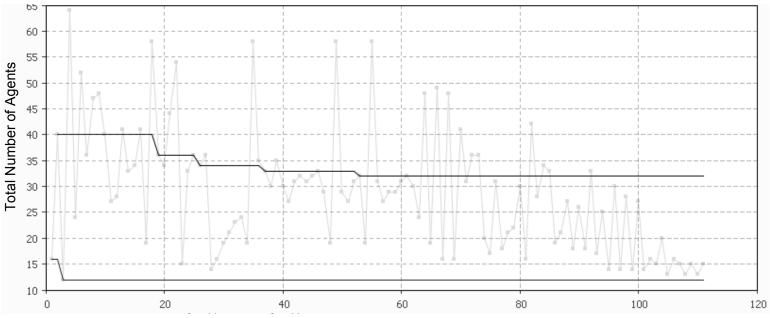

Figure 2: Convergence process of staffing configuration optimization.

4.2 Model Run

All of the above collected data are taken as input to the simulation model. We implement the simulation model using AnyLogic (XJ Technologies 2011), which serves as a user friendly interface to the modeling environment. The simulation of the functional unit is replicated multiple times to ensure that tight confidence bounds on the metrics are obtained. The model first simulates the current staffing levels to assess "as-is" service performance. Afterwards, the combined tabu search and scatter search methodologies are used to iteratively propose new staffing levels, and simulated to assess the "to-be" performance. This process is repeated until the stopping criterion is reached. Figure 2 shows the convergence process of the simulation-optimization framework, as implemented in AnyLogic. The x-axis indicates the number of iterations. The y-axis is the value of the objective function. A decrease trend is observed in the total value of the objective function as the simulation-optimization procedure searches for improved solutions in different iterations of the procedure. In the optimization phase, a solution is deemed feasible if the constraints that measure backlog stability and the constraints that measure service level attainment are satisfied. A solution is deemed optimal if it achieves minimal cost, where cost is measured by the total cost of the agents in the service delivery unit. We note that different costs can be assigned to agents with different skills, for example.

The suggested solution is then piloted in the functional unit for one month, during which the detailed performance data (e.g., service request backlog, service attainment levels) are gathered to assess the service performance. The pilot operates as follows. In the case that there is a recommendation to reduce the staffing levels in a functional unit, these agents are removed from the "production team" (which are servicing the customer service requests) and assigned to the "flexi team" where they are handling non customer related work. When at certain point the performance data indicate that the customer service level targets are at risk, an agent from the flexi team is called upon to work on the customer service requests and released when the risk has been remedied. The use of a flexi team provides a low-risk way to pilot the model recommendation and still maintain customer service level targets. Following the completion of the pilot period, the pilot results are analyzed to determine whether and to what extent the staffing levels suggested by the simulation model can be implemented.

4.3 Numerical Results

In this section we demonstrate the use of the simulation model via numerical results. The purpose of this study is to demonstrate how the simulation model can be used to (i) understand current system performance (*as-is performance*) and (ii) explore how changes in system configuration impact system performance. We measure system performance by the staffing costs, but model service level attainments as strict constraints (to reflect the reality in the service delivery environment).

Our study is based upon data from a large IT services delivery provider. We focus on a single service functional unit servicing five customers. The functional unit is comprised of two service delivery units. We refer to the more skilled unit as the expert ("*E*") team and the less skilled one as the basic ("*B*") team. There are four categories of customer requests handled by this functional unit: incidents, service

Table 1: Workload statistics.

	Incidents	Service Requests	Console Alerts	Projects
Avg. Weekly Volume	895	20	29,471	300
Avg. Handling Time (min.)	18.15	19.46	0.75	15.62 (*E*), 12.84 (*B*)
Stdev. Handling Tme	17.38	15.71	0.2	12.07 (*E*), 8.67 (*B*)
Target Response Time (min)	15	30	15	60
Pct. Attainment	97%	97%	97%	97%

requests, console alerts, and project requests. The requests differ in their arrival rate distributions, handling time distributions, and target response times. The console alerts constitute the highest workload (measured both in volume and in time expended), while the project requests can only be handled by agents in the more skilled *E* team and have a higher average response time. Table 1 lists the statistics for the base case scenario including average weekly volume of workload for the different service classes, average and standard deviation of handling time, target response time (time in queue), and percentage attainment.

There are 36 agents in this functional unit; 11 of them are on the *E* team and the remaining agents are members of the *B* team. The functional unit offers 24x7 coverage to all supported customers. There are a total of 9 shifts across the two service delivery units in this functional unit. Four of the shifts work in two pairs of two to provide 24 hour coverage as follows: In Shift 1, the agents work for three days on a 12 hour shift (6am-6pm), and then are off for three days. In the “paired shift” (Shift 2), the agents work on the same days, but the opposite 12 hours (6pm-6am). The second pair of shifts (Shift 3 and Shift 4) has a similar shift schedule, shifted by 3 days so that they are working on the three days that this first pair of shifts is off. Another shift in this functional unit provides service only during weekdays. The majority of the *B* shifts provide 24 hour support (rotating across the different agents associated with the shift); the main responsibility of these agents is to respond to the console alerts which have tight SLAs and require a quick response (within minutes). The majority of the *E* shifts do not provide 24 hour support. The primary responsibility of these agents is to handle the project requests whose associated contractual service level agreements only require support during normal customer business hours.

We first model this *base scenario*. The simulation results indicate that staffing levels can be reduced by 3 basic agents while still meeting all customer service quality metrics. We then systematically consider changes to various model inputs to measure the impact on required staffing levels while ensuring that service quality is met. In our first two experiments, we explore the impact of chages in the contractual terms of the service quality agreement. In Experiment (1) we modify the required percentage attainment from 97% to 95% attainment, representing a less stringent requirement on the monthly percentage attainment. This means that within each class of requests, 95% of the requests in the class must meet their associated service level target or, stated differently, if more than 5% of the requests in each class miss their target response time, the provider will be required to pay significant penalties. In Experiment (2), we modify the target times for the two most stringent request classes, Incidents and Console Alerts, from 15 minutes to 30 minutes. As anticipated, relaxing the terms of the service level agreements in these ways results in a reduction in the required staffing levels for the functional unit. The suggested reduction in both cases is targeted at the B service delivery unit. This too is reasonable; the B service delivery unit is responsible for the service requests that are most significantly impacted by modifications to terms in the service level agreements.

Besides staffing level recommendation, the simulation model can also be used to perform what-if analysis and explore the impact of service configuration changes. We demonstrate it through an example as shown in Figure 3. The *x*-axis indicates the five different service configuration experiment settings, and the *y*-axis indicates the minimum number of agents required in each setting so that the service quality metrics can be met. Experiment (1) is the as-is setting where 36 service agents (25 junior agents in the Rhythm unit and 11 senior agents in the Blues unit) are in the current service functional unit. Experiment (2) illustrates that only 33 agents are needed to satisfy the 97% service attainment target contracted with

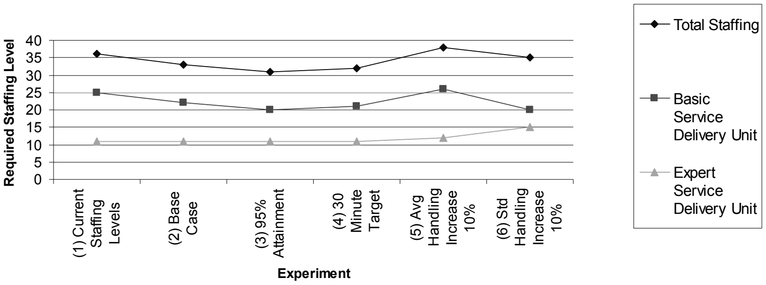

Figure 3: Minimum required staffing levels for different system configurations.

the customer. This means that within each class of requests, 95% of the requests in the class must meet their associated service level target or, stated differently, if more than 5% of the requests in each class miss their target response time, the provider will be required to pay significant penalties.

In experiment (3) we explore the impact of changes in the contractual terms, that is, we modify the required attainment percentage from 97% to 95%, representing a less stringent requirement on the monthly service level agreement. Thus results in two staff reduction compared to the to-be setting. Note that the suggested reduction is targeted at the Rhythm service delivery unit. This is because the console alert workload has the largest workload volume but requires basic skills for resolving. In experiment (4), we still explore the impact of contractual term changes, but modify the target times for the two most stringent request classes, Problems and Console Alerts, from 15 minutes to 30 minutes. This leads to one staff reduction compared to the to-be setting

Next, we modify the parameters of the handling time distribution and measure its impact. In experiment (5) we increase the average handling time in each class by 10%. This would be the case, for example, if new required steps are added to the procedures that must be followed by the agents in order to respond to requests. The simulation model suggests increased staffing is required. Further, the model suggests that staffing be increased in both service delivery units. Increased average handling time impacts all service request classes; the Console Alert request class is primarily handled by the *B* service delivery unit whereas the Project request class is primarily handled by the *E* service delivery unit explaining the suggested increase in both service delivery units. The model suggests a more significant increase in the *B* service delivery unit versus the *E* service delivery unit (an increase of 4 *B* and 1 *E* agents). This is commensurate with the volume of workload observed by each of these service delivery units. Comparing the cost associated with the increased required staffing against the changes that could possibly be implemented in order to reduce the average handling time can help the service delivery provider determine where dollars can be best spent. In experiment (6), we increase the standard deviation of the handling time in each service class by 10%. Here again the model suggests that increased staffing is required. However, in this case the impact is to the number of agents in the more skilled *E* service delivery unit. This can be explained by the base case standard deviations for the relative volumes of workload in each service class for which each service delivery unit has primary responsibility. Agents in the *B* service delivery unit have primary responsibility for the Console Alert request category which has high volume. On the other hand, the service time standard deviation for this request category is low, so a 10% increase has minimal impact. The remaining categories of service requests (primarily handled by agents in the *E* service delivery unit) have significantly higher standard deviations explaining the greater impact of this experiment on the number of required *E* agents.

5 CONCLUSIONS AND FUTURE WORK

The services delivery business is highly dynamic and highly competitive, with thin profit margins. Strict service quality targets coupled with highly variable service request arrival patterns and ever increasing cost containment targets make it challenging for a service delivery provider to deliver consistent quality and remain profitable. Due to various Lean initiatives, there is little room for the provider to pilot alternative solutions that may or may not result in improvements in system performance (reduced cost, improved service quality, etc.) A service modeling framework provides a solution to this challenge. The simulation model is a proxy for the real world business environment. This framework can be used to baseline as-is system performance by measuring the current staffing levels that are in place in each service delivery unit and evaluating the ability of the current staffing levels to meet service quality objectives. The modeling framework can then be used to perform various what-if analysis. For example, a services delivery provider can utilize this modeling framework to measure the impact of adding or removing customers from a service functional unit. Or, it can be used to evaluate the trade-offs of agent training to increase the skill profile versus hiring new agents with a limited set of specific skills.

While the initial results are encouraging, there are various challenges ahead of us. For example, one challenge that we face with our approach is the dynamic nature of the services delivery environment. The supported customer base as well as the number of agents experience frequent changes. Consequently, by the time the modeling results are available (after all data cleaning and analysis), the environment may have changed. Our response to this challenge was to implement methods to shorten the modeling cycle time from initial data collection to completion of final results. We are also studying which features of the model are most sensitive to changes in the input data.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to Brian Eck, Dipyaman Banerjee, Gargi B Dasgupta, Nirmit V Desai, and Rodney Wallace, all employed by IBM, for helpful and constructive discussions that helped us improve the quality of the model. In addition, we are indebted to Anatoly Zherebtsov, employed by XJ Technologies, for his assistance in model development.

REFERENCES

Aksin, Z., M. Armony, and V. Mehrotra. 2007. “The Modern Call Center: A Multi-Disciplinary Perspective on Operations Management Research”. *Production and Operations Management* 16:665–688.

Anerousis, N., Y. Diao, and A. Heching. 2010. “Elements of System Design Optimization in Service Quality Management”. In *Proceedings of IFIP/IEEE Network Operations and Management Symposium*, 48–55.

Atlason, J., M. A. Epelman, and S. G. Henderson. 2008. “Optimizing Call Center Staffing Using Simulation and Analytic Center Cutting-Plane Methods”. *Managment Science* 54:295–309.

Bouzada, M. A. C. 2009. “Scenario Analysis within a Call Center Using Simulation”. *Journal of Operations and Supply Chain Management* 2:89 – 103.

Brown, L., N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao. 2005. “Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective”. *Journal of the American Statistical Association* 100:36–50.

Cezik, M. T., and P. LaEcuyer. 2008. “Staffing Multiskill Call Centers via Linear Programming and Simulation”. *Management Science* 54:310–323.

Feldman, Z., and A. Mandelbaum. 2010, December. “Using Simulation Based Stochastic Approximation to Optimize Staffing of Systems with Skills Based Routing”. In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 3307–3317. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gans, N., G. Koole, and A. Mandelbaum. 2003. “Telephone call centers: Tutorial, review, and research prospects”. *Management Science* 5:79–141.

Glover, F. 1977. “Tabu Search”. *Decision Sciences* 8:156–166.

Glover, F., J. Kelly, and M. Laguna. 1999, December. “New Advances for Wedding Optimization and Simulation”. In *Proceedings of the 1999 Winter Simulation Conference*, edited by P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. Evans, 255–260. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Glover, F., and M. Laguna. 1977. “Heuristics for Integer Programming Using Surrogate Constraints”. *Decision Sciences* 8:156–166.

Office of Government Commerce 2007. “IT Infrastructure Library. ITIL Service Support, version 3, http://www.itil-officialsite.com/”.

Robbins, T. R., and T. P. Harrison. 2008, December. “A Simulation Based Scheduling Moel for Call Centers with Uncertain Arrival Rates”. In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 2884–2890. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Ulrich, R., and J. Miller. 1993. “Information processing models generating lognormally distributed reaction times”. *Journal of Mathematical Psychology* 37:513–525.

XJ Technologies 2011. “http://www.xjtek.com/”.

AUTHOR BIOGRAPHIES

YIXIN DIAO is a Research Staff Member at the IBM Thomas J Watson Research Center in Hawthorne, New York. He received his Ph.D. degree in Electrical Engineering from Ohio State University in 2000. He has published more than sixty papers in systems and services management and is the co-author of the book ”Feedback Control of Computing Systems” (Wiley 2004). He received IBM Outstanding Innovation Award in 2005 and was named to IBM Master Inventor in 2007. He is the recipient of the 2002 Best Paper Award at IEEE/IFIP Network Operations and Management Symposium, the 2002-2005 Theory Paper Prize from the International Federation of Automatic Control, and the 2008 Best Paper Award at IEEE International Conference on Services Computing. He is a Senior Member of IEEE, an Associate Editor for Journal of Network and Service Management, a Steering Committee Member for FeBID, and a Program Committee Member for various management conferences including IM, NOMS, DSOM, CNSM, and ICAC. He was workshop co-organizers for FeBID 2007 and AMACS 2007, and Program Co-chair of 2010 International Conference on Network and Service Management. His email address is diao@us.ibm.com.

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 modeling, simulation, statistical analysis, and design of decision support tools for system management and optimization in the areas of pricing, risk management, and workforce management. Her email address is ahechi@us.ibm.com.

DAVID NORTHCUTT is an IBM Distinguished Engineer with over 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 (Northwestern), M.S. in Computer Science (Univ. of Illinois at Chicago), and an M.S. in Statistics (Rutgers). He is currently a member of the Delivery Technology and Engineering organization in IBM where he leads and consults with IBM service delivery teams and IBM clients worldwide. He is an American Society for Quality (ASQ) Certified Quality Engineer and a Senior Member of ASQ. His email address is northcutt@us.ibm.com.

GEORGE STARK is an IBM Senior Technical Staff member with over 25 years of experience in software and service measurement and statistical modeling. He has published more than 40 technical papers in referred journals and conferences and has been on the editorial board of the Software Quality Journal as well as past chair of ISSRE. George is currently a member of the Delivery Excellence team where he consults with IBM quality and productivity improvement teams worldwide and is a key leader in the IBM Estimation Community of Practice. He also works with clients on project estimation, software reliability and process improvement approaches. His email address is gstark@us.ibm.com.