Utilizing the Waiting-time Criterion for Selecting Services in a Composition Scenario

Abhishek Srivastava, Paul G. Sorenson
Department of Computing Science
University of Alberta
Edmonton, Canada
{*sr16, paul.sorenson*}*@ualberta.ca*

***Abstract*—Service composition is an effective practice to perform complex tasks through varied configurations of simple services. An issue that often arises in such a set-up is the selection of the best service from a group of functionally equivalent ones to cater to each of the various functionalities of the composite application. Previous effort in this direction incorporates utilizing the Quality of Service (QoS) attributes of the services to pick out the best one of the lot. This paper presents a technique to select the optimal service for composition using the average waiting time attribute of the services. The service selected is the one that has the smallest value of the average waiting time. Concepts from queueing theory have been borrowed and customized to estimate the waiting time values of the candidate services. Experiments have been performed wherein selection results using the proposed technique are compared with the selections that are made by simulating an actual scenario and computing the waiting time by observation.**

I. Introduction

Service composition refers to the practice of forming composite applications to get some useful work done using groups of simple service components [1]. The constituent service components each perform their respective tasks, and putting these together a larger more complex task gets done. A common example of service composition is the 'trip-planner' application. The trip-planner application is a composite application which may comprise a flight-booking service, a hotel-booking service, a taxi-booking service, an integrated payment service *etc.* Each of the service components does a simpler task and together the more complex task of planning all aspects of the trip gets done.

With most countries in the world swiftly moving towards a service based economy [2] [3], service composition is becoming a more common practice by the day. This is a logical development as service composition enables effective re-use of existing service components. Customers stand to gain in this set-up in terms of the increased choices of applications at their disposal and have the liberty to demand 'tailor-made' applications to exactly cater to their respective requirements. Service providers also gain by virtue of the 'malleability' that the service composition practice lends to the process of service delivery.

In spite of the aforementioned advantages, the service composition practice is not without its issues. One of the issues that is dealt with in our research, is that of selecting the best components for the task at hand. With a large number of service components being available, every task that needs to be performed as part of the composite application has a number of candidate services. For example, in the trip-planner application mentioned earlier, the task of flight-booking could be performed by any of a number of services, *viz.* 'Expedia' [4], 'travelocity' [5], 'Flight Centre' [6]. Selecting the best service from the available functionally equivalent candidate services is non-trivial. Attempts in the past at selecting the best service from a set of functionally equivalent ones have been made utilizing the 'Quality of Service' (QoS) attributes of the services in question [7], [8], [9]. The services in question are ranked on the basis of one or more QoS attributes using various procedures and the highest ranked service component is selected for the respective task.

In this paper, the 'average waiting time' attribute of the service components is utilized to select the best service for the task. The average waiting time of a service is the time that a request that is being serviced by the same has to wait on an average before being serviced. The waiting has to be done because the service is busy servicing another request and possibly other requests waiting in a queue before the request in question. The smaller the average waiting time, the more desirable is the service. The *best* service among the functionally equivalent candidate services is therefore the one with the smallest value of the average waiting time.

To compare service components on the basis of their respective average waiting time values, we borrow simple concepts from Queueing Theory [10]. Approximate waiting time values are obtained for the candidate service components for each task, and the one with the smallest waiting time value is selected. An important point to note is that the queueing theory concepts used in this work have been derived under the assumption that the queueing system is in a 'steady-state'. The steady-state in a queueing system is a condition when the probability of the queue having a certain number of requests is constant. Attaining the steady-state in a dynamic service composition environment is however usually not possible. Therefore, the concepts borrowed have been customized to cater to a non steady-state environment.

The remainder of this paper is structured as follows. Section 2 is the related work section where papers that have presented techniques to minimize waiting-time in the past have been discussed. These techniques are not necessarily in the context of service composition. Section 3 is a description of the service domain that has been used in our research, within which the service composition takes place. Section 4 presents the proposed technique for calculating the average waiting time of the candidate services. It first describes the queueing theory concepts borrowed and subsequently describes the customization of the borrowed concepts to cater to the service domain being used. Section 5 consists of a description of the experiments conducted to validate the proposed technique and also consists of some of the results of the experiments. Finally, section 6 concludes the paper.

II. Related work

The related work included in this section is not necessarily in the context of service composition. However, all the work discussed here is related to the waiting-time criterion being used for service selection and performance modification.

Ismail *et al.* [11] tackle the issue of reducing the waiting-time of a collaborative project by selection of collaborating partners in a manner that the idle time of one coincides with that of the other. They explore patterns in the usage of the partners such that situations where one partner is occupied in another application while a second partner is idle is avoided. The technique is relevant only in a more static set-up where the commitments of the collaborating organizations are more defined. This technique may not be suitable in a more dynamic environment like that of service composition.

Wang *et al.* [12] present a technique which also borrows concepts from queueing theory to calculate the "expected" waiting time of services in a composition set-up. However, while doing this they make the simplifying assumption that the service composition queue is in the 'steady-state', by assuming that the request 'arrival-rate' is always smaller than the service 'completion-rate'. This however is not always true. Second, their method of calculating the request arrival-rate is based on observing the behavior of each service component for a long time. This is usually not practicable owing to the dynamism associated with the characteristics of each service in a composition environment.

Zuhair *et al.* [13] utilize the average waiting-time criterion in the context of an elevator system to modify the "jerk" and "acceleration" of elevators. The calls on the various floors are observed and if the difference between the current average waiting-time and the regular average waiting time exceeds a certain "unacceptable delta", the jerk and acceleration of the elevators is increased. The jerk and acceleration are brought down, once the average waiting-time goes down.

Flockhart *et al.* [14] present a technique to minimize the waiting-time for calls in a call-centre set up. The technique involves forming 'agent-queues' for each skill. Whenever a call is received requiring a certain skill, the agents in the queue corresponding to the skill are looked up. The number of queues that each of these agents is present in is computed. The agent that is present in the smallest number of queues is the one selected. This ensures that the waiting time for subsequent calls is minimized. In a service composition set-up, this technique could be utilized in a situation where each service element is capable of performing more than one task. Whenever a request for a task comes along, the service that is available, capable of performing this task, and which is capable of performing the minimum number of other tasks amongst all available services is selected.

Green [15] utilizes queueing theory in healthcare in the allocation of resources in an effective manner. The complexity and unpredictability of a hospital set-up notwithstanding, Green shows the effectiveness of an $M/M/s$ queueing model in the allocation of resources such as beds and staff to minimize the waiting-time of patients as well as enabling optimal utilization of resources. The demand flexibility in hospitals is dealt with by application of the queueing model over smaller 'staffing periods' rather than over the whole day.

III. The Service Domain

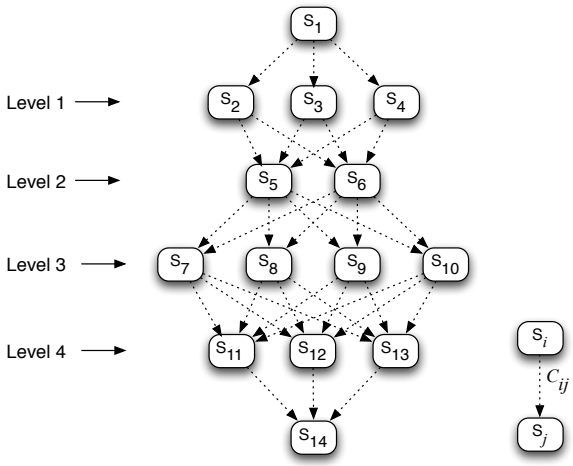

Figure 1. The service domain with arcs representing coupling

The service domain in our work has been represented as a set of levels, with each level corresponding to a unit functionality of the composite application. Each of the levels is instantiated by the group of service components that are capable of catering to the functionality represented by the respective level. From each of the levels, one service needs to be selected to be part of the composite application. The levels are arranged from top to bottom in such a way that the first service to be invoked is the top most one and this service invokes one of the services at the next immediate

level and so on. In an attempt to keep the model simple, the order of service invocation is from top to bottom and never from bottom to top, thus avoiding cycles. In a situation that a functionality is to be performed several times, the level is repeated several times in the domain. Figure 1 shows the service domain representation followed in our work. Services S_2, S_3, and S_4 are functionally equivalent. Similarly S_5, and S_6 are functionally equivalent and so on. For example, Figure 2 is the representation of a possible service domain of a trip-planner application. The top most level of this domain includes the 'log-in' procedure, authentication *etc.* The next level corresponds to the flight-booking functionality. The services populating this level are each capable of booking a flight for the customer. One of these needs to be selected for the application. The subsequent functionality levels in the trip-planner application domain are respectively from top to bottom, the taxi-booking functionality, the hotel-booking functionality, the combined payment functionality (hypothetical service examples have been used), and finally the 'log-out' procedure.

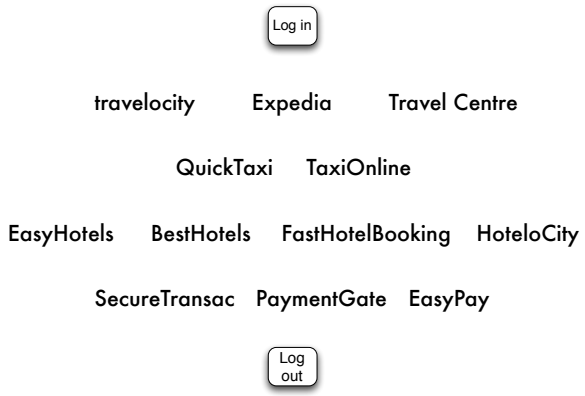

Figure 2. The service domain of the trip-planner application

Further in the service domain representation, a factor called *Coupling* joins each service at a certain functionality level in the domain to every service at the next lower functionality level. The coupling C_{ij} between a service i at a given level and a service j at the next lower level is a dimension-less factor which expresses the likelihood, that given that service i is currently being used to cater to the functionality at its level, the next service to be invoked will be j to cater to the next level of functionality. The higher the value of the coupling, the higher is this likelihood. The coupling is represented by dotted arcs in the service domain representation shown in Figure 1.

The coupling depends on factors such as business relations between the respective service providers, and low level factors such as technical compatibility between the services. For example, in the trip-planner application of Figure 2, let the coupling between services *Expedia* and *QuickTaxi* be 2.4 and the coupling between *Expedia* and *TaxiOnline* be 4.3. Now every time that *Expedia* is invoked to carry out the flight booking functionality, there is a much larger possibility that *TaxiOnline* will be invoked to do the taxi booking rather than *QuickTaxi*.

IV. The Proposed Technique

The proposed technique for calculating the average waiting-time at each service involves employing concepts from queueing theory. The queueing theory concepts borrowed however are those meant for a 'steady-state' condition when the probability of the queueing system having a certain number of units is constant [10]. In a service composition environment, however, the steady-state condition is seldom attained. The queueing theory concepts therefore need to be customized slightly to be utilized for the purpose of service composition. Besides this, customization also needs to be done to adapt to the specific nature of the service domain used in our research (described in the previous section).

A. Average waiting-time in queueing theory

The queueing system that has been utilized is the $M/M/1$ queue. This is the most basic queue, where the first M stands for negative exponential time distribution of request arrivals, the second M stands for negative exponential distribution of service completion, and the 1 stands for the fact that there is just one queue (no parallel queues). Figure 3 shows the $M/M/1$ queue where λ denotes the request 'arrival-rate' and μ denotes the service 'completion-rate'.

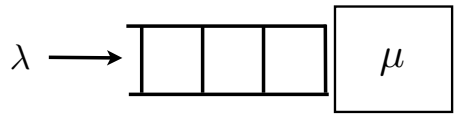

Figure 3. The basic $M/M/1$ queue

This is the most common kind of queue which is observed in most everyday activities like the customer queues in banks, movie-halls *etc.* A queue like this is characterized by the number of units present within the queueing system. For example, in the queue in a bank, if the number of people waiting in the queue is 5, the total number of people in the queueing system would be 6, *i.e.* $(5+1)$, the 5 people waiting and the 1 person being served. The 'state' of that queueing system at that point of time is said to be 6. A queueing system of this kind, for a short period of time has a randomly varying number of states. To characterize the behavior of the queue during short time spans is very challenging. However, if the system is observed over a large span of time, it often attains a 'steady-state' wherein the probability of the system being in any state is found to be constant. For larger time spans, the queue behavior is more predictable and useful conclusions may be drawn for the same. To attain the steady-state however, one basic requirement is that the request

arrival-rate should be smaller than the service completion-rate of the system ($\lambda < \mu$). If this condition is not satisfied, the number of waiting units will progressively increase and the steady-state would never be reached.

In the steady-state situation, the average waiting-time for requests in an $M/M/1$ queue is given by the expression in equation (1) [10]. In this equation, as the value of λ increases (arrival-rate increases), the value of $(\mu - \lambda)$ in the denominator decreases. The increasing λ in the numerator and the decreasing $(\mu - \lambda)$ in the denominator cause the average waiting-time to increase. This makes intuitive sense, as the arrival-rate and hence number of arrivals increases in a queue, the average waiting-time increases. However, this makes sense as long as λ is less than μ. As soon as the two become equal, the average waiting-time becomes ∞, and if λ increases beyond μ, the average waiting-time becomes negative.

$$\textit{Average waiting time} = \frac{\lambda}{\mu \cdot (\mu - \lambda)} \tag{1}$$

B. Customization of average waiting-time for the composition scenario

The average waiting-time expression borrowed from the queueing system described in the previous sub-section needs to be customized for the composition scenario for two reasons: 1) the request arrival to a service may be from any of a number of services at the functionality level immediately above, thus the arrival-rate would need to be appropriately modified; 2) the dynamic nature of the service composition environment does not allow the attainment of a steady-state which is a necessary requirement for using the average waiting-time expression in equation (1). This is because it cannot be assured in the dynamic service composition environment that the request arrival-rate (λ) is always smaller than the service completion-rate (μ).

To determine the request arrival-rate for a service in a multi-level domain, a simple case is first considered. Assume the service domain at each level comprises of only one service. The request arrival-rate at a service l would then simply be equal to the completion-rate of the service k immediately above it (ignoring other latency factors). This is because, every request on being served by k would enter the queue of service l.

For a service domain with multiple candidate services at each functionality level, a request may arrive from any of the services from the level immediately above it. The 'coupling' values described in the previous section are used in determining which candidate service is chosen by the request. Recall that coupling is a characteristic feature between two services i and j at adjacent functionality levels such that a higher coupling value C_{ij} results in a greater likelihood of service i invoking service j. This means that a request on being completed by a service at one level would next move to a service at a lower level which has a higher value of coupling with the current service. The request arrival-rate at the queue of the lower level services therefore depends on the values of coupling between the lower level services and the upper level services and on the completion-rate values of the services at the upper level.

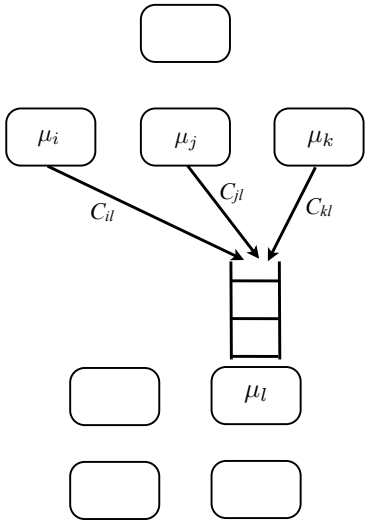

Figure 4. The arrival-rate derived from completion rate of ancestor services

Figure 4 explains this. Service l is a service at a certain level which has coupling values C_{il}, C_{jl}, and C_{kl} with services i, j, and k respectively at the level directly above it. Services i, j, and k respectively have completion rate values equal to μ_i, μ_j, and μ_k. The value of the request arrival-rate for service l derived from these completion rate values is shown in equation (2).

$$\textit{Arrival-rate}_l(\lambda_l) = \frac{C_{il}}{C_{il} + C_{jl} + C_{kl}} \cdot \mu_i + \frac{C_{jl}}{C_{il} + C_{jl} + C_{kl}} \cdot \mu_j + \frac{C_{kl}}{C_{il} + C_{jl} + C_{kl}} \cdot \mu_k \tag{2}$$

The second customization that needs to be done to borrow the concepts of the $M/M/1$ queueing system is to recognize that the steady state condition is difficult to be attained in a composition environment owing to the dynamism associated with a composition scenario. This is due to the fact that it is difficult to ensure that the request arrival-rate at each service is always less than its completion-rate. Several 'trial and error' experiments were performed to modify the average waiting-time expression of equation (1), and finally the average waiting-time expression was modified as shown in equation (3) to give an approximate figure of the average waiting-time of a queue in conditions that did not qualify as 'steady-state'. It may be argued that such a modification of the average waiting time expression could deem the latter invalid. We however conducted experiments, results of which

are provided in the next section, that establish its validity even in conditions that do not qualify as steady state. Moreover, the modified expression also makes sense intuitively. As the request arrival-rate λ increases for a constant μ, the value of the average waiting-time increases according to equation (3). This seems logical because an increase in the arrival-rate of requests at a queue for a constant completion-rate would result in a stacking of requests and thus a higher waiting-time. Similarly, according to equation (3) as the completion-rate μ increases for a constant λ, the average waiting-time decreases, because the μ is in the denominator of the expression. This also seems fair because if the completion-rate of the service increases for a constant arrival-rate, the service would start processing the requests much faster and there would be a resulting fall in waiting-time. Furthermore, this expression does not have the requirement of $\lambda < \mu$. Even when λ is greater than μ, it still makes sense. Thus, the steady-state restriction of the original Queueing Theory waiting-time expression is overcome.

$$\textit{Average waiting time} = \frac{\lambda}{\mu \cdot (\mu - \lambda)} \Rightarrow \frac{\lambda}{\mu} \tag{3}$$

V. Experimental Validation

Experiments were conducted on a service domain with 29 services (excluding the first and the last) spread over 6 levels of functionality. The experimental domain is shown in Figure 5.

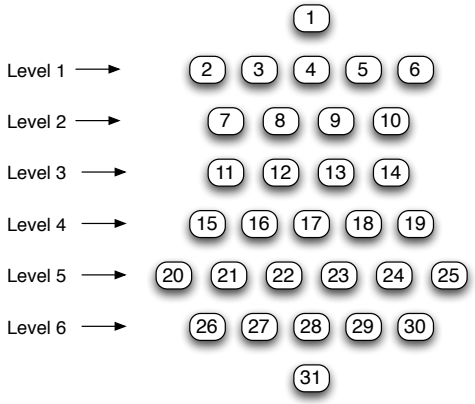

Figure 5. Service domain used in the experiments

The experimental procedure comprised simulating the behavior of the domain for a large number of service requests, and calculating the average waiting-time at each service by observing the time spent by the requests on an average in the service queue. Next, the proposed technique was applied on the domain and the waiting-time calculated at each service. Although, numerically the waiting-time calculated through simulation and the proposed technique were found to differ (this could be owing to variable simulation parameters), the service ranking in terms of waiting-time was found to be almost identical in either procedure, at each level.

A. Simulation

The simulation procedure is described in a little detail in the following portion. Seven different sets of completion-rate values and coupling values were experimented with. Each experimental set also had an application request arrival-rate which was substantially larger than any of the completion rate values. $100,000$ events were allowed to happen randomly at the assigned rates. An event comprised either the arrival of a new request for the entire composite application or the completion of any of the services, servicing a request. The progress of the simulation is traced as follows: an application request that arrives at the top of the domain immediately enters service 1 (Figure 5) to be serviced if the latter is idle or joins the queue if service 1 is busy servicing another request. Once this request is serviced by service 1, it has to move to one of the services at the next level. The service to which this request moves depends upon the coupling values between the services at the next level and service 1, and the number of requests already waiting in the respective queues of the services at the next level. The decision on which service to move to is on the basis of equation (4) and is illustrated in Figure 6. The service selected in Figure 6 is the shaded one which although has a smaller coupling value, has a smaller number of services in its queue.

$$\textit{next-service}_i = max(\frac{coupling_{ij}}{1 + \textit{number of units}_j}) \tag{4}$$

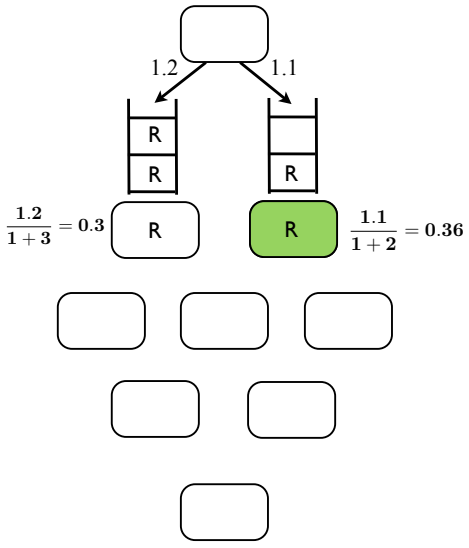

Figure 6. The movement of requests from one level to another during simulation

The process of moving from one functionality level to the next continues for the request until it reaches the lower-most

concluding level. This process is repeated for a large number of requests, and while the simulation is being run, the time spent by the requests in the queues of each of the services it goes through is noted. Finally, the average waiting-time for each of the services is calculated by dividing the total waiting-time of all the requests that were serviced by it, with the number of requests serviced.

The motivation for running the simulations was to get an idea of how a domain (if one existed) behaved in terms of waiting-time if it were observed for a large period of time (one that allowed $100,000$ events).

B. Proposed technique application

The proposed technique comprised of simply calculating the request arrival-rate values for each service in the domain using the expression in equation (2) first. Subsequently, the calculated arrival-rate values were substituted in the average waiting-time expression in equation (3). With the completion-rate values already given, the average waiting time value for each service was calculated.

C. Results

The results of the experiments for two sets of completion-rate and coupling values are shown in Figures 7, and 8.

In Figures 7, and 8 as mentioned before, although the numerical values of the waiting-time calculated during simulations and using the proposed technique do not match, the important point to look out for are the service ranks that have been allotted to the services on the basis of the average waiting-time values calculated (smaller waiting-time services are more highly ranked). The rankings have been shown on the right-hand side of the tables in the two figures. The ranks allotted by each technique are found to be in almost perfect conformance.

The proposed technique is therefore found to almost perfectly match the selections of the simulation (since the rankings of the services determines the selection).

VI. Conclusion

This paper presents a technique to utilize the average waiting time to select a service from a group of functionally equivalent ones in a service composition scenario. The selected service is the one that has the smallest waiting time value. Queueing theory concepts have been borrowed to give a rough estimate of the waiting time values. The concepts borrowed are intended for use under a steady state condition. However, attaining the steady state in a dynamic service composition scenario is rare, hence the queueing theory concepts have been appropriately customized.

The advantage of this technique is that it enables a rough estimation of the waiting time through the application of elementary formulae. The technique is validated by comparing the proposed technique results with the waiting time values calculated by observation in a simulated environment. It was found that in almost all cases the service ranking on the basis of waiting-time using the proposed technique almost exactly matched the observed values.

References

[1] Shahram Dustdar and Mike P. Papazoglou, *Services and Service Composition An Introduction*, it-Information Technology, Volume 50, 2008.

[2] Patrizia Battilani and Francesca Fauri, *The rise of a service-based economy and its transformation: the case of Rimini*, Rimini Centre for Economic Analysis, Working Paper Series, 2007.

[3] Faiz Gallouj, *Innovation in the service economy: the new wealth of nations*, Edward Elgar Publishing, 2002.

[4] *Expedia*, http://www.expedia.com

[5] *travelocity*, http://www.travelocity.com

[6] *FlightCentre*, http://www.flightcentre.com

[7] Shuping Ran, *A model for web services discovery with QoS*, ACM SIGecom Exchanges, pp. 1-10, 2003.

[8] Natallia Kokash, *Web service dicovery with implicit QoS filtering*, Proceedings of the IBM PhD Student Symposium, in conjunction with the International Conference on Service Oriented Computing (ICSOC), pp. 61-66, 2005.

[9] Zhengdong Gao and Gengfeng Wu, *Combining Qos-based service selection with performance prediction*, IEEE International Conference on e-Business Engineering (ICEBE), pp. 611-614, 2005.

[10] Donald Gross and Carl M. Harris, *Fundamentals of Queueing Theory*, Wiley Series in Probability and Statistics, 1998.

[11] Azlan Ismail, Jun Yan, and Jun Shen, *Dynamic Service Selection for Service Composition with Time Constraints*, Proceedings of the Australian Software Engineering Conference, 2009.

[12] Xiaoling Wang, Kun Yue, Joshua Zhexue Huang, and Aoying Zhou, *Service Selection in Dynamic Demand-Driven Web Services*, Proceedings of the International Conference on Web Services, 2004.

[13] Zuhair S. Bahjat and Gerald B. Fried, *Automatic selection of different motion prole parameters based on average waiting time*, United States Patent 5290976.

[14] Andrew D. Flockhart, Robin Harris Foster, Roy A. Jensen, Joylee E. Kohler, and Eugene P. Mathews, *Call center agent selection that optimizes call wait times*, United States Patent 6192122.

[15] Linda Green, *Queueing Analysis in Healthcare*, International Series in Operations Research and Management Science, 2006.

Domain 1		Average waiting-time values						Service rank (smaller waiting-time ahead)					
Level 1	Simulation	2.9	0.9	1.6	0.7	1.1		5	2	4	1	3	
	Proposed	208.3	65.79	122	57.8	99.01		5	2	4	1	3	
Level 2	Simulation	5.3	1.7	2.2	1.3			4	2	3	1		
	Proposed	0.511	0.173	0.223	0.145			4	2	3	1		
Level 3	Simulation	39.9	58.1	173.1	29			2	3	4	1		
	Proposed	0.246	0.316	0.918	0.206			2	3	4	1		
Level 4	Simulation	0.5	1.0	1.2	0.5	0.8		1	4	5	1	3	
	Proposed	0.095	0.213	0.245	0.096	0.156		1	4	5	2	3	
Level 5	Simulation	0.5	0.8	1.3	0.5	1.3	0.6	1	4	5	1	5	3
	Proposed	0.143	0.234	0.374	0.146	0.4117	0.187	1	4	5	2	6	3
Level 6	Simulation	0.6	0.5	3.1	0.5	1.1		3	1	5	1	4	
	Proposed	0.137	0.125	0.65	0.125	0.234		3	2	5	1	4	

Figure 7. Level wise comparison of results between simulation and the proposed technique (Domain 1)

Domain 2		Average waiting-time values						Service rank (smaller waiting-time ahead)					
Level 1	Simulation	0.9	0.4	6.7	0.6	1.7		3	1	5	2	4	
	Proposed	78.125	36.764	454.55	49.26	140.85		3	1	5	2	4	
Level 2	Simulation	0.7	0.6	0.6	1.4			3	1	1	4		
	Proposed	0.1517	0.1504	0.1317	0.3087			3	2	1	4		
Level 3	Simulation	0.9	0.7	1.4	0.5			3	2	4	1		
	Proposed	0.2966	0.2325	0.4269	0.1641			3	2	4	1		
Level 4	Simulation	0.5	0.7	0.8	0.6	0.4		2	4	5	3	1	
	Proposed	0.2041	0.2418	0.3405	0.2301	0.1728		2	4	5	3	1	
Level 5	Simulation	1.1	1.0	0.6	0.4	0.7	1.1	5	4	2	1	3	5
	Proposed	0.3557	0.3414	0.2040	0.1289	0.2313	0.3848	5	4	2	1	3	6
Level 6	Simulation	0.8	0.5	0.7	1.6	0.4		4	2	3	5	1	
	Proposed	0.1865	0.1313	0.1787	0.3615	0.1020		4	2	3	5	1	

Figure 8. Level wise comparison of results between simulation and the proposed technique (Domain 2)