Proposed Models for predicting Tangerine Call Centre Performance

Thomas Eapen **thomas@nowlabs.net**

The activities based configuration of the Genesys platform along with the complexities of its forecasting module have resulted in us looking at alternative models for forecasting major Key Performance Indicators (KPIs) such as Average Speed of Answer (ASA) and Service Level (SL). In addition to predicting Call Centre performance, we are looking at models for calculating the number of Agents required to meet our target KPIs.

To begin the process of modelling said data, we first realized that we had to get around the broken activity model that was generating incorrect coverage data and head count. The decision was made in consultation with the Resource Planning Team (RPA) to generate a new tree hierarchy that was based on Team, Department and Language. These three dimensions allows us to bucket Agents into their location, trained capabilities and language skills.This simulation of Agent properties map each Associate to one major product line of the company. This simplistic model does not represent our multi skilled Agent capabilities however our skilling model (through levelling and protective thresholds) does bias Agents to focus on their primarily capability which translates to one major product line of the company. Hence we begin by assuming that the three dimensional bucketing of Agents is a fair approximation of the Agent's realistic workflow as opposed to workflow around his or her wholistic capability.

With the new three dimensional hierarchy in place, the next step involved generating Full Time Equivalent (FTE) hours per interval that takes into account all scheduled state groups that result in Associates being in a Not Ready or Logged out state. Our Shrinkage calculator that was built to generate this data also includes a snapshotting feature where shrinkage until the day before is snapshotted and compared to the actual shrinkage on the day off or after which provides additional insight into performance characteristics of the day. Timeframes for our shrinkage calculator can be honed down to 1 minute intervals and also takes into effect shifts that overlap day boundaries. These features allow us to calculate accurate Scheduled Hours for a specific timeframe down to Location, Department, Language filters. The Shrinkage calculator does not currently look at the productivity factor which takes into account time spent out of adherence that negatively affects scheduled hours. The plan is to add this feature into the shrinkage calculator.

With the Shrinkage calculator built and giving us accurate Scheduled Hours information, we obtained forecasted workload information from the Forecasting Team. We looked at best practise modelling for Call Centre resource allocation and performance prediction. The first system for traffic design and analysis that we investigated was the Erlang C formula. We could have use one of the many free Erlang C calculators available on the web but we decided to implement the formula with Excel's built in formulas. The advantage of generating the formulae from scratch allowed us to see how input variables such as call arrival rate, call duration and number of agents affected parameters such as traffic intensity and agent occupancy. The Erlang C formula uses the poisson distribution function of Excel in the probability mass function mode and the cumulative density function mode to calculate the probability of a call waiting. The formula was implemented using the Excel scripting language and the Python programming language to check for accuracy.

The python code to implement Erlang C is below:

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jun  7 19:57:11 2016

@author: thomas
"""

from scipy.stats import poisson
from math import exp

#Call Volume
cvl = 360
#in seconds
interval = 1800
#average arrival rate
h = cvl / interval
#average call duration
t_s = 240
#number of agents
m = 55
#traffic intensity
u = h * t_s
#agent occupancy
p = u / m

def erlangc(m, u):
    numerator = poisson.pmf(m, u)
    denominator = numerator + ((1 - u/m) * poisson.cdf(m-1, u))
    return numerator / denominator

print("Probability of waiting: {e_c:2.1%}".format(e_c=erlangc(m, u)))

#Average Wait Time (ASA)
asa = (erlangc(m, u) * t_s) / (m * (1 - p))

print("ASA: {0:.2f}s".format(asa))

#Target ASA
t = 15

#Service Level
sl = 1 - erlangc(m, u) * exp(-(m - u) * t / t_s)

print("Service Level: {0:2.1%}".format(sl))

####Output

Probability of waiting: 23.9%
ASA: 8.18s
Service Level: 84.6%
```

The implementing of Erlang C with Python also showed some of the limitations of the model. Primarily if the agent occupancy is greater than 1, then the agents are overloaded and the Erlang C calculations are not meaningful[1].

The other reasons for caution are:

- The Erlang B and C traffic models make certain assumptions about the nature of the call arrivals. Amongst them is the assumption that call arrivals are random (Poisson arrivals). Although this is quite reasonable in most applications, it can cause inaccurate results when there is a sudden peak of calls[2].
- The Erlang C traffic model does not take abandoned calls into account, but if your call centre is engineered correctly, this should not be a factor. It may cause a problem when attempting to use Erlang C to analyze an existing call centre with poor performance[3].

The Erlang C model also does not take multi skilled Agent capabilities into effect however the assumption is that skilling bias described earlier allows us to use this model with reasonable Accuracy. With the shrinkage calculator providing us scheduled hours for the major product lines and with awareness of the limitations of the Erlang C model, we proceeded to model daily performance using freely available Erlang C add ons for Excel[4]. The different KPI forecasts such as ASA, Abandoned Rates and SL provided general insight into how the queues would perform with inconsistent spikes that occurred in the occasional interval. When compared to the actual performance characteristics for the day, it was obvious that while the models generally predicted “good” or “bad” performance, good performance predictors were overly optimistic and bad performance predictors were exaggerated. We feel that the potential reason for this result is that the multi skilled agent capability of our centre is more pronounced with our skilling model than assumed. While we get general performance characteristics, we are unable to currently gauge a fair approximation of the number of Agents required when we input our target KPIs into the Erlang C formula pool.

The next steps that are being planned are

1. To simulate multi skilled FTE hours by looking at historical ratios of types of calls taken by multi skilled Agents and adjust FTE per product line based on those ratios.
2. Additionally, the productivity factor will be built into the Shrinkage Calculator to close any gaps in shrinkage calculations.
3. Erlang C calculations can be a surprisingly complicated series of steps and recommendations will be made to the department heads on buying a supported product with good supporting documentation rather than using one of the freely available Erlang C calculators available
4. Investigate additional workforce planning models from sources such as the Call Centre Helper website[5].

---

[1] http://mitan.co.uk/erlang/elgcmath.htm

[2] http://erlang.com/calldesign.html

[3] http://erlang.com/calldesign.html

[4] http://www.erlang.co.uk/excel.htm

[5] https://www.callcentrehelper.com/images/forecasting-and-planning-multi-skilled-workforce-maggie-klenke.pdf