Erlang-A Simulation Model

By Nikolay Anisimov

July 7, 2010

Summary
It is well known that Erlang C and Erlang B models play an important role in telephony call centers as they allow estimating workforce demand for inbound and outbound calls. At the same time for real-world call centers these modes are quite limited as they do not take into consideration abandonment of calls. In operational research the queue model that takes into consideration abandoned calls is known as Erlang A. The model assumes all processes have an exponential distributions and has an analytical solution.

In this short document we describe a simulation model of Erlang A. The model describes a basic process and could be served as starting point for more sophisticated call center models such as multi-campaign environment, outbound dialing, multi-skilled agents, etc.

The model is created with the aid of simulation tool AnyLogic 6 and runs as a Java applet.

RUN Erlang-A Model!

Table of Contents

<table>
<thead>
<tr>
<th>Summary</th>
<th></th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Introduction</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Model</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Overall Model</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Model Pane</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Dashboard</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Control Pane</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Conclusion</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>6</td>
</tr>
</tbody>
</table>
**Introduction**

It is well known that Erlang C and Erlang B models play an important role in telephony call centers as they allow estimating workforce demand for inbound and outbound calls. At the same time, for real-world call centers these modes are quite limited as they do not take into consideration abandonment of calls.

Fortunately, the operation research community has extended Erlang model to so-called Erlang A model that is sometimes referred to as $M/M/n + M$ [1]. There is a strict analytical solution of the model that allows calculating main parameters of the system. However, they are quite complicated (for example, it uses incomplete Gamma function) for practical usage and becomes completely intractable when we deviate from exponential distribution functions to more realistic ones.

In this notes we consider a basic simulation model of Erlang A. We believe the model could be useful for understanding dynamic processes in call centers. The model is created with the aid of simulation tool AnyLogic 6 [2] and its run-time instance could be exported as Java applet.

Based on this simple model one can create more sophisticated models including multi-skilled routing, outbound dialing campaigns, interactions of other channels, etc.

**Model**

**Overall Model**

The overall model view is presented in Figure below. It contains three window panes:

![Model View](image)

**Figure 1: Model**
Model pane, Dashboard pane, and Control pane

The simulation window has a control bar on the top:

![Control bar image]

Using this bar one can control the execution of simulation model. For instance one can start, stop and resume execution of the model as well as its speeding-up and slowing-down.

In next sections all these panes will be considered in details.

**Model Pane**

Structure of the model is represented in bottom right window pane. It is composed of the following elements:

- Inbound source element that generates inbound calls from customers. We will assume that the flow is Poisson arrival.
- Service element that contains an inbound queue and a processing block. When a call enters the service element it is placed at the end of the queue if all agents are busy. If however there are available agents, the call immediately proceeds to an agent without any waiting.
- Agent pool contains a group of agents dedicated for call handling. The model assumes that call handling time has an exponential distribution.
- Sink element for calls processed by agents. When an agent finished handling of the call, the call leaves the system via the sink.
- Sink element for calls that left the queue due to abandonment by customers. When a call stays in the queue waiting for available agent, the customer may decide to leave the system by hanging on a receiver. The corresponding call leaves the system and considered to be abandoned. The model assumes that the waiting time to abandonment has an exponential distribution.

![Model pane diagram]

Figure 2: Model pane
On left side model contains four variables with self-explained names. Actual run-time values of the variables are represented under the variable names (in blue).

Each element is accompanied with a numeric value that varies in a run time. For instance from this picture we could say that

- 499 calls were generated by inbound source element which were proceeded to the queue of the service element;
- Currently the queue contains 45 calls;
- Currently 165 calls are handled by agents in service element;
- Zero agents are free;
- 28 calls left the queue due to abandonment;
- 261 calls were processed by agents and left the system via sink element.

**Dashboard**

Dashboard pane shows a set of current service indicators:

- Service Level defined as a fraction of calls (in %%) which handling were started in 20 seconds.
- Abandonment Rate (AR) defined as percentage of calls abandoned by customers;
- Agent Occupancy (AO) calculated as a percentage of busy agents to all agents;
- Average Speed of Answer (ASA) – an average waiting time for calls reached agents (not abandoned)
- Queue length representing number of calls waiting in a queue.

Dashboard also contains “Inbound Occupancy” calculated with the formula

\[
\frac{arrival\ rate \times AHT}{number\ of\ agents} \times 100\%
\]

Notice this value may exceed 100% that corresponds to understaffed regime and high abandonment rate.

**Figure 3: Dashboard pane**

Dashboard pane also has some dynamic plots and chat object.
• Agent occupancy represented by pie and plot charts.
• Service level is represented by third pie chart. It contains three segments: green part corresponds to fraction of calls which processing started within 20 seconds (so here service level is equal to 93.3%). Red part represents part of calls abandoned from the queue (here abandonment rate is equal to 3.5%). Yellow part is a fraction of calls processed by agents but waited in the queue more than 20 seconds.
• Queue length is represented by stack chart. In normal well-staffed regime the length is close to zero.
• Reset button allows resetting calculation of service indicators.

Control Pane
Control pane is represented in Figure below. It allows changing main parameters of the system using four sliders.

![Control pane](image)

The model has following parameter that could be changed in a run time:

• Arrival rate defined as an average number of call arrivals in one second
• Average handling time (AHT) in seconds
• Number of agents
• Average time to abandonment (ATA) in seconds.

One can play with the sliders to change system parameters and see how it influences the system behavior.

Conclusion
In this notes we described a simulation model of Erlang A built with the aid of AnyLogic 6. It should be stressed that the model is too ideal and differs what happens in reality. The main deviation is a handling time distribution. In real-world call centers handling time has a log-normal distribution. Fortunately, AnyLogic tool allows using a wide spectrum of distribution function including log-normal one.
The model also could be extended in many ways including multi-skilled agents and outbound dialing campaigns.

**RUN Erlang-A Model!**

**References**
