

Análisis y Diseño de Call Centers

Germán Capdehourat
Evaluación de Performance de Redes de Telecomunicaciones
Trabajo Final 2006

Resumen

Este trabajo presenta una parte de la investigación académica realizada hasta la fecha en materia de centros de atención telefónicos, más conocidos como call centers [1][2]. La mayor parte de la misma tiene su origen o se basa en la teoría de colas. Sin dudas, esta perspectiva de los call centers es natural y además útil para el análisis y diseño de estos sistemas. Dichos modelos han servido para generar herramientas estándar que apoyan la gestión de los call centers. Sin embargo, los call centers modernos son sistemas cada vez más complejos, con características nuevas como IVR¹, ruteo basado en habilidades, chat, e-mail, etc. que hacen más difícil el análisis y sobrepasan los límites de la teoría de colas existente al momento. Aparece entonces la simulación como única vía para el manejo de estos casos de extrema complejidad.



Figura 1. Ejemplos de call centers modernos.

1. Introducción

Debido a los grandes avances en las tecnologías de la información, el número, tamaño y alcance de los call centers, así como la cantidad de gente trabajando en ellos o utilizando sus servicios como clientes, ha crecido en gran forma durante la última década. A modo de ejemplo, sólo en EEUU, la industria de los call centers se estima que emplea varios millones de personas como agentes, sobrepasando la agricultura [3]. En Europa, el número de empleados de call centers fue estimado entre 1999 y 2000, indicando por ejemplo, 600.000 en el Reino Unido (2.3 % del total de trabajadores), 200.000 en Holanda (casi 3 %) y entre 300.000 y 400.000 en Alemania (1-2 %). Una nueva disciplina denominada Ingeniería de Servicio [4], que busca dar soporte al diseño y gestión de las operaciones involucradas en los servicios, tiene entre sus campos de acción a los call centers.

1.1. Modelo general

Un call center [1][2] consiste en un complejo sistema, tanto desde el punto de vista tecnológico, como en lo que refiere a la interacción humana. La figura 2 muestra un modelo general de lo que puede ser un call center con un solo tipo de llamada entrante.

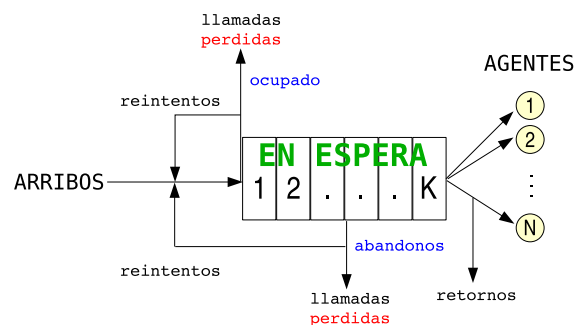


Figura 2. Modelo general de un call center.

En el mismo se puede ver un sistema con N agentes atendiendo llamadas y $N + K$ líneas telefónicas (con $K \geq 0$). Un cliente cuando llama tiene tres resultados posibles:

¹Interactive Voice Response

1. Ser atendido inmediatamente si hay algún agente libre.
2. Esperar en cola, si todos los agentes están ocupados y hay líneas libres.
3. Ser bloqueado, por no haber líneas disponibles.

Luego, para el caso del cliente que queda en cola existen dos posibilidades:

1. Esperar hasta ser atendido, cuando algún agente se libere.
2. Abandonar el sistema sin ser atendido.

Además existen los reintentos, tanto para el caso de bloqueos como de abandonos, siendo estos los clientes que vuelven a llamar. Por otro lado se tienen los retornos, que corresponden a clientes que fueron atendidos pero por alguna razón, vuelven a comunicarse con el *call center*.

1.2. Indicadores de performance

A continuación se presenta una lista con los principales indicadores utilizados para analizar la performance de un *call center*. Se considera la variable aleatoria W , que mide el tiempo de espera de los clientes para ser atendidos.

Grado de servicio

Es el indicador de desempeño principal, desde el punto de vista del cliente. En general se lo denomina SL , por su nombre en inglés *Service Level*. Corresponde al porcentaje de llamadas que son atendidas antes de una cierta demora fija, parámetro que se elige según la calidad de servicio que se quiere brindar.

$$SL = \frac{N^{\circ} \text{ de llamadas atendidas antes de } AWT}{N^{\circ} \text{ de llamadas atendidas}}$$

siendo el AWT ² el tiempo medio de espera aceptable. En términos estocásticos, el grado de servicio es:

$$SL = P(W < AWT)$$

Espera media

Es el tiempo medio que espera un cliente en ser atendido. Se denomina en general como ASA , sigla en inglés de *Average Speed of Answer*.

$$ASA = E\{W\}$$

²Acceptable Waiting Time

Porcentaje de abandonos

Corresponde a la fracción de llamadas del total que arriban, en que los clientes abandonan por agotar su paciencia.

$$P(Ab) = \frac{N^{\circ} \text{ de abandonos}}{N^{\circ} \text{ total de arribos}}$$

Una variante de este indicador es contabilizar solamente aquellos abandonos que aguardan un tiempo mínimo (ϵ pequeño), eliminando de esta forma a los clientes ansiosos.

$$P(Ab|W > \epsilon) = \frac{N^{\circ} \text{ de abandonos que esperan mas de } \epsilon}{N^{\circ} \text{ total de arribos}}$$

Porcentaje de bloqueos

Es la fracción de llamadas del total que arriban, que son bloqueadas por no tener líneas disponibles.

$$P(Blk) = \frac{N^{\circ} \text{ de bloqueos}}{N^{\circ} \text{ total de arribos}}$$

Porcentaje de ocupación de los agentes

Corresponde a la fracción de tiempo del total que opera el *call center*, en que los agentes están atendiendo llamadas.

$$P(Blk) = \frac{\sum^{AGENTES} \text{Tiempo atendiendo llamadas}}{\text{Tiempo total de trabajo del call center}}$$

1.3. Preguntas a responder

En base a los indicadores de performance definidos, se establecen los objetivos de diseño del *call center*. Surgen entonces, diversos problemas asociados que se deben resolver. A partir de un objetivo de diseño, en el cual se define el grado de servicio, la probabilidad de bloqueo, el porcentaje de ocupación de los agentes, etc. lo que uno desea es hallar los recursos necesarios para cumplir con los objetivos. Esto en la práctica involucra varios problemas, algunos de los cuales se listan a continuación:

Predicción de los arribos Se debe modelar los procesos de arribo, para poder estimar la carga a manejar.

Estudio del comportamiento humano Es necesario modelar los tiempos de servicio, la paciencia de los clientes, los reintentos, todos eventos asociados al comportamiento de las personas.

Modelos estacionarios Se consideran intervalos donde se cumpla esta propiedad, por lo que es necesario estudiar el funcionamiento del sistema, fraccionando el día en períodos donde se cumpla dicha hipótesis.

Esquema de horarios Luego de hallar los agentes para cada intervalo, se debe armar un esquema de horarios compatible para tener la cantidad necesaria de agentes en cada período, con jornadas de trabajo razonables, ya sea de 6 horas, 8 horas, etc.

2. Modelos basados en teoría de colas

Como se menciona al comienzo, el estudio de los *call centers* está basado principalmente en la teoría de colas, donde se utiliza un proceso de nacimiento y muerte para modelar la cola de espera del *call center*. Los parámetros del sistema se consideran constantes durante cierto período (ej: media hora), para trabajar con un modelo estacionario. A partir del modelo, se obtiene la distribución estacionaria del sistema y con la misma es posible hallar en forma analítica los indicadores del desempeño del *call center*. A continuación se presenta dos modelos que siguen este procedimiento.

2.1. Erlang-C

El modelo tradicional para el dimensionado de *call centers* es el M/M/N, siendo N la cantidad de agentes y el número de líneas ∞ . Este modelo considera arribos y tiempos de servicio exponenciales de tasas λ y μ respectivamente. No se toma en cuenta las demás posibilidades presentadas como abandonos, bloqueos, reintentos, etc.

En la figura 3 se puede ver tanto el modelo de Erlang-C, como la cadena asociada al mismo. A partir de este modelo, podemos llegar a una expresión analítica para el SL y el ASA, en función de N, λ , μ .

Siendo $E_c(N, \rho) = \frac{\rho^N}{(N-1)!(N-\rho)} \frac{1}{\frac{\rho^N}{(N-1)!(N-\rho)} + \sum_{j=0}^{N-1} \frac{\rho^j}{j!}}$ donde $\rho = \lambda/\mu$ quedan:

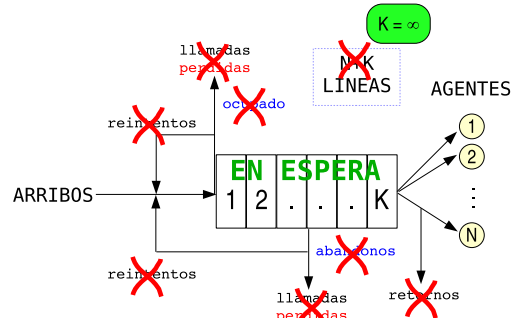
$$SL = 1 - E_c(N, \rho) \times e^{-(N-\rho) \cdot AWT \cdot \mu}$$

$$ASA = \frac{E_c(N, \rho)}{(N - \rho) \cdot \mu}$$

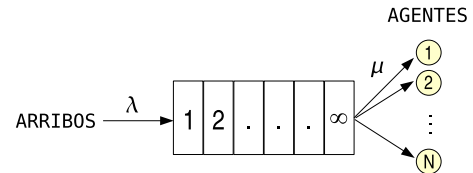
2.2. Erlang-A: Considerando abandonos

Este modelo [3], es una extensión del anterior y se basa en los resultados de Palm [5]. Se denomina M/M/N+M, puesto que se modela la paciencia de los clientes mediante una variable aleatoria exponencial, donde los abandonos se producen a tasa θ . Esto significa que el tiempo medio que un cliente espera en ser atendido es $1/\theta$. En [3] se argumenta la importancia de considerar los abandonos en el modelo.

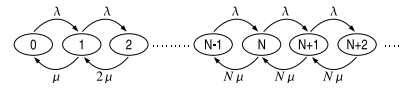
Como se observa en la figura 4, la cadena asociada al modelo de Erlang-A difiere de la de Erlang-C cuando se



(a) Simplificaciones al modelo general.



(b) Modelo simplificado resultante.



(c) Cadena asociada a la cola de espera.

Figura 3. Modelo de *call center* Erlang-C.

tienen por lo menos N+1 clientes en el sistema, es decir a partir de que la cola no está vacía. También es posible en este caso hallar la solución analítica, pero resulta más complejo que para Erlang-C. Por más detalle ver [3].

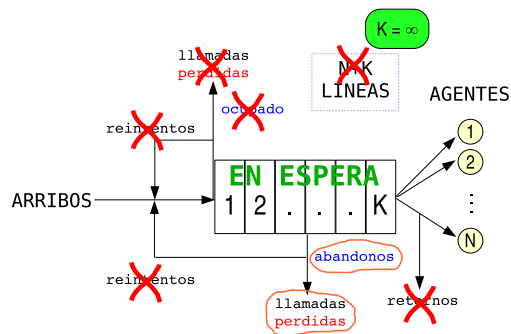
2.3. Comparación de Erlang-C y Erlang-A

En el siguiente ejemplo se realiza una comparación entre ambos modelos presentados para un sistema particular. Los parámetros utilizados son:

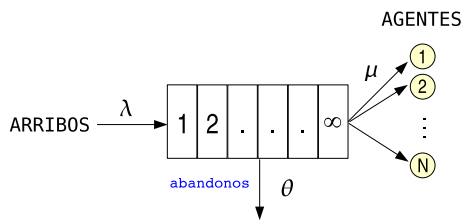
- $\lambda = 50 \text{ min}^{-1}$ (50 llamadas por minuto)
- $\mu = 1/30 \text{ seg}^{-1}$ (2 servicios por minuto)
- $\theta = 1 \text{ min}^{-1}$ (1 abandono por minuto de espera)

Se releva los indicadores $P_{\text{ESPERAR}} = P(W > 0)$, $SL = P(W < 20 \text{ seg.})$ y $ASA = E(W)$, todos ellos para un rango del número de agentes, que varía desde 15 a 45. Se ve claramente la diferencia entre ambos modelos, en particular Erlang-C es inestable para $N < 26$, mientras que Erlang-A es siempre estable en el intervalo mostrado.

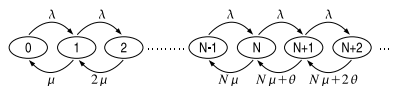
Para el caso de particular de $N = 26$ agentes, se presenta la tabla 2.3. En la misma podemos ver que para la misma tasa de arribos, el tamaño medio de la cola y la espera promedio es bastante menor para el caso de Erlang-A, con una ocupación de los agentes un tanto menor y tan solo un 4,9 %



(a) Simplificaciones al modelo general.



(b) Modelo simplificado resultante.



(c) Cadena asociada a la cola de espera.

Figura 4. Modelo de call center Erlang-A.

de abandonos. Este pequeño porcentaje de abandonos se refleja en una calidad de servicio muy superior para los clientes que *sobreviven*. La razón es que los abandonos reducen la carga justo cuando se necesita, es decir cuando la congestión es alta. Además, podemos ver que para el caso de Erlang-C, si disminuimos la tasa de arribos en igual medida que el porcentaje de abandonos, de todas formas obtenemos una performance por debajo que para el caso de Erlang-A. Podríamos llegar a la misma performance aumentando la cantidad de agentes, pero esto es justamente lo que queremos minimizar puesto que es el mayor costo del *call center*.

	Erlang _C	Erlang _A	Erlang _C $\lambda \downarrow 4,9\%$
Abandono (%)	-	4.9	-
Espera Media (seg.)	23.5	2.9	7.5
Cola Promedio	19.6	2.5	6.0
Ocupación (%)	96.2	91.4	91.4

Este modelo, justifica el hecho comprobado empíricamente de que muchas veces un diseño determinístico es una buena solución. Para este caso particular, el diseño sería $N = 25$, los cuales atienden 2 llamadas por minuto y cubren las 50 llamadas por minuto que llegan. El desempeño para este caso es muy bueno, con $P_{ESPERAR} = 39,36\%$,

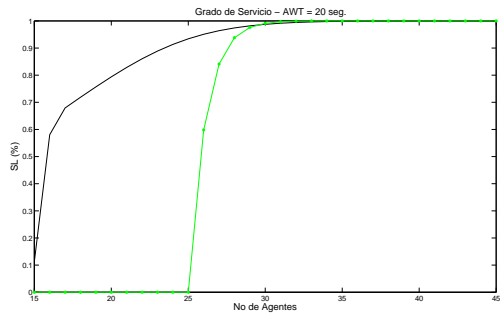
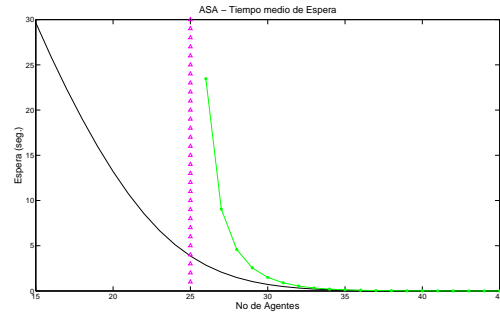
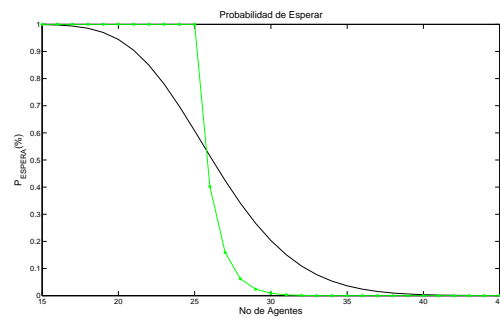


Figura 5. Comparación de los modelos de Erlang-A y Erlang-C.

$SL_{20 \text{ seg.}} = 93,4\%$ y $ASA = 3,86 \text{ seg.}$, cuando para el modelo tradicional de Erlang-C el sistema sería inestable.

3. Regímenes de operación

Uno de los objetivos en el diseño y gestión de un servicio en general, y en el caso de un *call center* en particular, es lograr un balance entre la calidad de servicio que se brinda y la eficiencia con que se utilizan los recursos. Considerando la elección del número *correcto* de agentes, esto se traduce en no contratar de más, para evitar el sobredimensionamiento, ni de menos, puesto que se vería afectada la calidad de servicio. Según la aplicación específica de cada *call center*, el diseño estará orientado a optimizar el sistema desde el punto de vista del cliente o desde el punto de vista de la eficiencia en la ocupación de los agentes. Surgen del análisis, distintos regímenes asintóticos de interés que simplifi-

can el cálculo analítico de los indicadores de performance. Los mismos se presentan brevemente a continuación, en este caso en el marco del modelo de Erlang-A. En todos los casos N es el número de agentes y $\rho = \lambda/\mu$.

3.1. ED - Efficiency Driven

El diseño está orientado a la eficiencia, por lo que se busca trabajar cerca de un 100% de ocupación de los agentes. En este caso virtualmente todos los clientes esperan. Este regimen es útil para aplicaciones sin fines de lucro (ej: donaciones), donde se quiere sacar el máximo provecho a los recursos. En este regimen la regla de diseño es:

$$N = \rho \cdot (1 - \gamma) + o(\sqrt{\rho}), \quad \gamma > 0$$

donde la fracción de abandonos en este caso tiende a γ y el tiempo medio de espera es aproximadamente λ/θ .

3.2. QD - Quality Driven

El diseño está orientado a la calidad de servicio. Se utiliza para aplicaciones donde la misma es mucho más importante que la eficiencia, como puede ser el caso de clientes VIP o teléfonos de emergencia. La regla de diseño queda:

$$N = \rho \cdot (1 + \gamma) + o(\sqrt{\rho}), \quad \gamma > 0$$

La espera media, el porcentaje de abandonos y la probabilidad de esperar tienden a 0 exponencialmente con N .

3.3. QED - Quality & Efficiency Driven

En este caso se tiene un compromiso entre la calidad de servicio y la eficiencia. Se busca dar buena calidad de servicio pero con un alto grado de ocupación de los agentes. Es generalmente el regimen de operación de centros de ventas, atención al cliente, etc. Este regimen se diseña con la denominada *regla de la raíz cuadrada*³:

$$N = \rho + \beta\sqrt{\rho} + o(\sqrt{\rho}), \quad -\infty < \beta < \infty$$

donde β es un parámetro que define la calidad de servicio. Se observa que para porcentajes de abandono moderados se cumple $-1 \leq \beta \leq 2$.

4. Ruteo basado en habilidades

En los modelos presentados de Erlang-C y Erlang-A, ambos consideran un solo tipo de llamada entrante. A continuación se presenta el caso más complejo de un *call center* que recibe varios tipos de llamadas. En este caso se define además, varios grupos de agentes, cada uno de ellos con

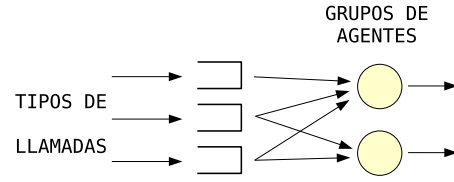


Figura 6. Call center con ruteo basado en habilidades.

distintas *habilidades*, las que definen qué tipo de llamadas puede atender cada uno.

Las tecnologías de ruteo basado en habilidades, conocido en inglés como *skill-based routing* (SBR), permiten implementar la diferenciación de distintos tipos de llamadas/clientes y varios grupos de agentes. La selección de los tipos de llamadas/clientes es una tarea de marketing, mientras que la definición de los grupos de agentes es una tarea de gestión de recursos humanos. El correcto mapeo de los grupos con las necesidades consituye un gran desafío de diseño. Es necesario decidir en tiempo real la elección de la llamada a atender así como el agente al que se le asigna y además qué llamada atiende de la cola, un agente que se libera.

4.1. Matriz agentes-habilidades

Una de las decisiones a tomar es definir qué tipo de llamadas sabe atender cada grupo de agentes. Una opción posible para esto es utilizar la denominada matriz agentes-habilidades [6], en la cuál se definen tanto las habilidades de cada agente como la prioridad para atender cada una de ellas. En esta matriz A , de tamaño $N \times C$, siendo N el número de agentes y C los tipos de llamadas, las filas corresponden a los agentes mientras que las columnas corresponden a las prioridades. Si la entrada $A_{i,j} = h$ esto significa que el agente i tiene la habilidad h con nivel de prioridad j ; si $A_{i,j} = 0$, entonces el agente i no tiene la habilidad h con nivel de prioridad j ; si no existe j tal que $A_{i,j} = h$, entonces no puede atender dicho tipo de llamadas. Por lo tanto, la fila i define las habilidades y sus prioridades para el i -ésimo agente. Se asume en este caso que un agente tiene a lo sumo una habilidad para una prioridad dada (no tendría por qué ser así, ver 6.1) y que para cada agente, una habilidad dada puede estar a lo sumo en un nivel de prioridad determinado. La primera columna de A tiene las prioridades primarias, es decir las más altas para cada agente. Por lo tanto $A_{i,1}$ es la habilidad primaria para el agente i . Se asume que cada agente tiene una habilidad primaria.

³Square Root Staffing Rule

Luego de definidos los agentes y las habilidades, es necesario además, elegir una política de ruteo. En particular se debe especificar las decisiones a tomar en dos casos:

1. Cuando llega una llamada.
2. Cuando un agente queda libre.

En el primer caso, cuando se tiene una nueva llamada entrante del tipo h , la misma se rutea al grupo de agentes que tienen esta habilidad con nivel de prioridad 1. Para asignarla a un agente particular existen varias opciones, destacándose la denominada LIAR⁴, por ser justa en el reparto de carga para los agentes. Esta política asigna la llamada al agente que hace más tiempo se encuentra disponible. Si todos los agentes que tienen la habilidad h como primaria se encuentran ocupados, se repite el proceso con los que la tienen como secundaria y así sucesivamente hasta encontrar un agente libre o en su defecto, si no hay agente libre que pueda atender dicha llamada, encolar la misma.

Para el caso en que un agente queda libre, si no hay llamadas en la cola, el mismo queda libre. De lo contrario, el agente busca en la cola alguna llamada de las habilidades que posee. La búsqueda se realiza según el orden de prioridad de las habilidades que tiene, esto es, primero la habilidad primaria, luego la secundaria y así hasta encontrar una llamada para atender. De no haber una llamada que pueda atender, también queda libre. Esto se puede ver como una recorrida de las colas de cada habilidad h , según el orden de prioridad del agente, donde para cada cola el servicio es FIFO, también denominado FCFS⁵.

4.2. Matriz de prioridades

Otra manera de definir la asignación de agentes a cada tipo de llamada, es a través de la matriz de prioridades [7]. Esta matriz es de igual tamaño que la de agentes-habilidades, donde las filas corresponden también a los agentes pero a diferencia del otro caso las columnas corresponden a los tipos de llamadas y no a las prioridades. De esta forma, la entrada de la matriz $R_{i,k}$ indica el nivel de prioridad de la llamada de tipo k para el agente i . Si el nivel de prioridad toma valores de 1 a ∞ se debe elegir un valor especial (ej: 0) para el caso en que el agente no atienda ese tipo de llamadas.

Esta estructura permite la existencia de distintos tipos de llamadas con igual nivel de prioridad para un determinado agente. Por otro lado el ruteo se vuelve más complejo, debido a que cuando los niveles de prioridad son iguales, es necesario un segundo algoritmo para desempatar. Luego de definida la llamada a atender, la elección de un agente determinado es análoga a lo descrito anteriormente.

⁴Longest-Idle-Agent-Routing

⁵First-Come First-Served

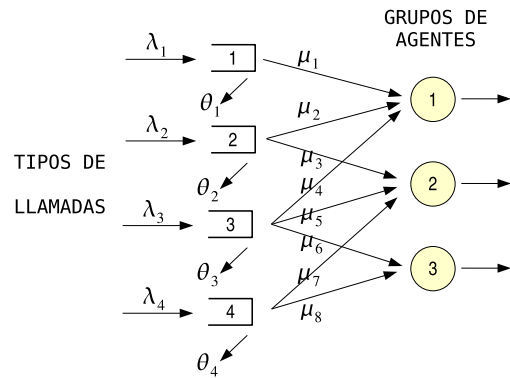


Figura 7. Modelo de *call center* con ruteo basado en habilidades.

4.3. Modelado del sistema

Estos sistemas son de mayor complejidad que el caso simple de un solo tipo de llamadas, por lo que el estudio analítico se vuelve extremadamente complejo. En la figura 7 el modelo de un caso particular, considerando colas según el modelo de Erlang-A. Existe intensa investigación en el área [2], pero de todas formas la complejidad de los sistemas actuales excede los límites de la teoría de hoy día. Esta razón hace que la manera de atacar el análisis de estos sistemas se divida básicamente en dos caminos:

- Analizar, apoyado en la teoría de colas, modelos simplificados del problema, aplicados al estudio de casos particulares y/o resultados asintóticos.
- Dejar de lado el estudio analítico y utilizar herramientas de simulación para analizar sistemas complejos.

4.4. Bloques canónicos

Una manera de las descritas para atacar los sistemas con SBR corresponde al análisis de casos específicos. En particular, en [8] se presenta distintos bloques canónicos, los cuales se pueden ver como piezas básicas para la construcción de un *call center* con SBR. En la figura 8 se presentan dichos bloques.

En [8] se menciona el hecho de que aún para estos diseños simples, el análisis se realiza mediante simulación en la mayoría de los casos puesto que el estado del arte del estudio analítico es aún insuficiente.

5. Simulación

La otra vía mencionada para el análisis de sistemas complejos corresponde a el uso de herramientas de simulación

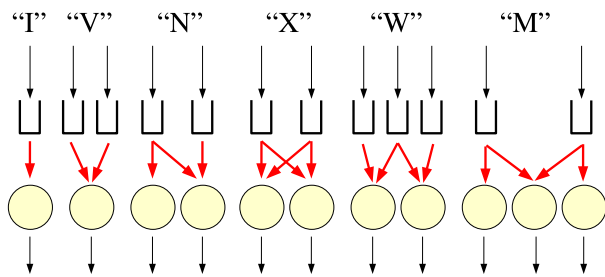


Figura 8. Bloques canónicos de un *call center* con ruteo basado en habilidades.

[9]. Este camino ofrece la ventaja de saltarse los límites que presenta el estudio analítico y analizar sistemas de extrema complejidad.

Existen diversas herramientas de simulación disponibles para el análisis de *call centers* [10, 11]. En este trabajo se presenta una librería de java denominada *Contactcenters* [7], mediante la que es posible simular *call centers* complejos y de esta manera realizar un análisis apropiado según el caso.

5.1. Arquitectura de la Librería *Contactcenters*

Contactcenters [7] es una librería de java que permite la simulación de complejos sistemas reales. El objetivo de la librería es brindar una plataforma para simular una gran variedad de *call centers*. Se busca una arquitectura flexible y con posibilidades de extensión, lo que se logra mediante la construcción de pequeños bloques independientes que se combinan según las necesidades. Los componentes elementales se describen a continuación.

Contactos

La clase *Contact* corresponde a cualquier tipo de contacto (llamada, mail, etc.) y tiene atributos como el tiempo de arribo, la prioridad, etc. que pueden accederse en cualquier momento de la simulación. Un contacto corresponde a un único cliente, pero un cliente podría necesitar hacer varios contactos antes de abandonar el sistema. También puede asociarse a un grupo de líneas, limitando de esta forma el acceso, puesto que si arriba un contacto cuando no hay recursos disponibles el mismo se bloquea.

Fuentes de tráfico

Las fuentes de tráfico determinan cuando deben crearse los contactos, según un cierto proceso estocástico de arribos. Este proceso puede depender del estado del sistema de manera compleja, pero en general sólo depende del tiempo de simulación y los tiempos de arribo previos.

Colas de espera

Las colas de espera (*WaitingQueue*) son estructuras cuyos elementos son *contactos* en espera. Para poder soportar abandonos, la cola tiene un calendario de eventos que ocurren para la eliminación automática de los contactos. Existen dos subclases, una estándar que corresponde a colas FIFO⁶ o LIFO⁷ y otra para colas con prioridades más complejas.

Grupos de agentes

Un grupo de agentes se representa como una instancia de la clase *AgentGroup* que registra el número de agentes libres y ocupados en cada instante de la simulación. El servicio de un contacto consta de dos pasos, el primero la atención propiamente dicha y el segundo asociado al trabajo posterior a la atención que el agente deba realizar, lo cual posibilita la simulación de diversas situaciones complejas. Además es posible simular eventos como que el router reciba una llamada e interrumpa a un agente que está respondiendo un mail para asignarle la misma.

Routers

Un router, comúnmente llamado distribuidor automático de llamadas (ACD⁸), corresponde a cualquier clase que se encargue de *escuchar* los nuevos contactos y asignarlos ya sea a un grupo de agentes o a una cola de espera. Además *escucha* los fines de servicio, para saber cuando un agente queda libre, y asignarle un nuevo contacto. Esta clase, a través de una subclase de la misma, define la política de ruteo con la que se trabaja. La librería trae algunas políticas predefinidas, pero es posible crear las propias, redefiniendo los métodos correspondientes de la subclase.

Otras consideraciones

Usualmente los modelos estacionarios de *call center* se utilizan para períodos de 15 a 60 minutos. Para realizar la simulación se definen P períodos en los que opera el *call center* y es necesario agregar dos períodos extra: uno al comienzo y otro al final. Esto es para que la simulación comience antes de que abra el *call center* (período preliminar) y termine luego de que el mismo cierre (período de cierre). Se debe definir además, *colectores* estadísticos que se encargan de recabar la información necesaria para calcular los indicadores de performance deseados.

⁶First-In First-Out

⁷Last-In First-Out

⁸Automatic Call Distributor

6. Call center bilingüe

La idea es estudiar mediante simulación un ejemplo concreto, en este caso un *call center* bilingüe. El mismo atiende dos tipos de llamadas, que podrían ser inglés y español por ejemplo, y tiene agentes unilingües quienes atienden llamadas de una sola lengua y agentes bilingües que atienden ambas. Este tipo de sistemas han sido estudiados de manera analítica [12] utilizando los denominados quasi-procesos de nacimiento y muerte. No pretende ser este un ejemplo de *call center* complejo, sino simplemente se utiliza para mostrar las capacidades de la simulación como herramienta de diseño.

6.1. Modelo

Como se observa en la figura 9, el modelo tiene arribos exponenciales de tasas iguales de valor λ para cada idioma. Los tiempos de servicio de los grupos de agentes son también exponenciales de tasas iguales de valor μ tanto para los especialistas (unilingües) como para los generalistas (bilingües). Las paciencias son también exponenciales de parámetros iguales de valor θ para ambos casos.

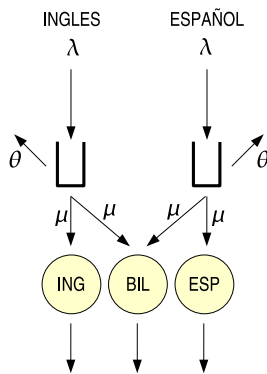


Figura 9. Modelo para *call center* bilingüe.

Si se considera el orden de izquierda a derecha del esquema 9 para numerar los grupos de agentes, es decir los que atienden sólo inglés el 1, bilingües el 2 y sólo español el 3, la matriz agentes-habilidades sería:

$$A = \begin{pmatrix} ING \\ ESP/ING \\ ESP \end{pmatrix}$$

Aquí las filas corresponden a los grupos de agentes y las columnas a las prioridades, por lo que hay una sola columna. Notar que en este caso la matriz no cumple las hipótesis planteadas en su definición, puesto que los agentes bilingües tienen igual prioridad para ambos tipos de llamadas, lo que impide utilizar la misma para definir la asignación de llamadas. Sin embargo, utilizando la matriz de prioridades

sí es posible hacerlo, quedando la misma de la siguiente forma:

$$R = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$$

Las filas corresponden a los grupos de agentes y las columnas corresponden a los tipos de llamadas, por lo que la prioridad 1 indica que atienden dichas llamadas y 0 que no lo hacen.

6.2. Simulación

La idea es estudiar el número de agentes necesarios de cada tipo para cumplir con ciertos requisitos definidos a priori. En este caso se busca cumplir con el siguiente grado de servicio $SL = P(W < AWT) > 80\%$ para una $AWT = 20$ segundos. Además, como se considera la posibilidad de abandonos, se exige que se cumpla $P(Ab) < 5\%$. Como se dijo anteriormente, el modelo no alcanza para tener definido el sistema, por lo que debemos elegir una política de ruteo para la simulación. En este caso se analiza las situaciones siguientes:

1. Las llamadas serán atendidas sólo por especialistas (unilingües), por lo que el número de agentes generalistas (bilingües) es 0.
2. Las llamadas serán atendidas sólo por generalistas (bilingües), por lo que el número de agentes especialistas (unilingües) es 0 para ambos idiomas.
3. Las llamadas serán ruteadas en forma aleatoria entre todos los agentes disponibles capaces de atender las mismas, es decir los especialistas del idioma correspondiente y los generalistas.

Los valores utilizados para la simulación son:

- $\lambda = 50 \text{ min}^{-1}$ (50 llamadas por minuto)
- $\mu = 1/30 \text{ seg}^{-1}$ (2 servicios por minuto)
- $\theta = 1 \text{ min}^{-1}$ (1 abandono por minuto de espera)
- Tiempo de simulación = 1 hora
- No de repeticiones = 1000

6.3. Resultados

En el caso de sólo agentes unilingües los resultados de la simulación son:

2× Agentes	SL (%)	Ocup. (%)	P(Ab) (%)
47	84.2	97.6	5.96
48	87.5	96.8	5.04
49	90.3	96.0	4.20
50	92.4	95.0	3.46

En el caso de sólo agentes bilingües los resultados de la simulación son:

1× Agentes	SL (%)	Ocup. (%)	P(Ab) (%)
93	91.5	99.3	5.42
94	92.5	99.0	4.86
95	93.4	98.7	4.33
96	94.2	98.3	3.83

En el caso de ambos tipos de agentes los resultados de la simulación son:

Agentes	SL (%)	Ocup. (%)	P(Ab) (%)
38/38/18	90.9	98.7	5.12
37/37/20	91.2	98.7	5.09
36/36/22	91.4	98.8	5.07
35/35/24	91.5	98.8	5.04
34/34/26	91.7	98.8	5.02
33/33/28	91.8	98.8	5.00
32/32/30	91.8	98.8	4.99
31/31/32	91.9	98.9	4.97
30/30/34	92.0	98.9	4.96

Fijando como objetivo minimizar la cantidad total de agentes a contratar, esto se logra claramente considerando todos los agentes bilingües. Puesto que los mismos constituyen el personal más capacitado, son más difíciles de conseguir. Por tal motivo se busca cumplir con los objetivos, minimizando el número de agentes bilingües. De la última tabla se desprende que el óptimo es 32/32/30. En una situación más real, es probable que los agentes bilingües cobren más que los unilingües, por lo que esto debe ser tenido en cuenta para minimizar el costo total en agentes.

7. Comentarios finales

Para concluir y a modo de resumen:

- Los *call centers* modernos son sistemas complejos, cuyo diseño y gestión constituyen un gran desafío y a la vez un área de investigación de mucho interés.
- El modelo de Erlang-A que considera abandonos, es una extensión del tradicional modelo de Erlang-C, utilizado para el estudio de *call centers*. Dada la importancia que tienen los abandonos en el desempeño de los *call centers* [3], se considera más adecuado trabajar con dicho modelo para un correcto dimensionamiento y análisis.

- Para el estudio de *call centers* de mayor complejidad es necesario dejar de lado el estudio analítico y utilizar herramientas de simulación para analizar dichos sistemas.
- La librería de Java *Contactcenters* [7] permite la posibilidad de simular este tipo de sistemas de alta complejidad, emulando el comportamiento real del *call center* en funcionamiento.
- Los *call centers* con ruteo basado en habilidades permiten sacar mayor provecho a los recursos. Como ejemplo se ve el aprovechamiento de la economía de escala aplicado a un *call center* bilingüe pero esto puede extenderse a una gran variedad de casos.

Referencias

- [1] Ger Koole. *Call Center Mathematics*. Department of Mathematics, Vrije Universiteit Amsterdam, Holland, Agosto 2006.
- [2] Noah Gans, Ger Koole, and Avishai Mandelbaum. Telephone Call Centers: Tutorial, Review, and Research Prospects. *Manufacturing & Service Operations Management*, 5:79–141, 2003.
- [3] Avishai Mandelbaum and Sergey Zeltyn. The Palm/Erlang-A queue, with applications to Call Centers. Faculty of Industrial Engineering & Management, Technion, Haifa, Israel, Junio 2005.
- [4] Avishai Mandelbaum and Sergey Zeltyn. Service Engineering of Call Centers: Research, Teaching and Practice. Faculty of Industrial Engineering & Management, Technion, Haifa, Israel, Junio 2006.
- [5] C. Palm. Research on telephone traffic carried out by full availability groups. 1957.
- [6] Rodney B. Wallace and Ward Whitt. A Staffing Algorithm for Call Centers. *Manufacturing & Service Operations Management*, 7(4):276–294, 2005.
- [7] Eric Buist and Pierre LÉcuyer. A java library for simulating contact centers. In *WSC '05: Proceedings of the 37th Winter Simulation Conference*, pages 556–565, Orlando, Florida, 2005.
- [8] Ofer Garnett and Avishai Mandelbaum. An Introduction to Skills-Based Routing and its Operational Complexities. Technion, Israel, Mayo 2000.
- [9] Vijay Mehrotra and Jason Fama. Call center simulations: call center simulation modeling: methods, challenges, and opportunities. In *WSC '03: Proceedings of the 35th Winter Simulation Conference*, pages 135–143, New Orleans, Louisiana, 2003.

- [10] V. Bapat. The Arena product family: Enterprise modeling solutions. In *WSC '03: Proceedings of the 35th Winter Simulation Conference*, pages 210–217, New Orleans, Louisiana, 2003.
- [11] NovaSim. ccProphet - simulate your call center's performance. Available online via www.novasim.com/CCProphet/.
- [12] D.A. Stanford and W.K. Grassmann. Bilingual Server Call Centres. In *Analysis of Communications Networks: Call Centres, Traffic and Performance*, volume 28, pages 31–48. Fields Institute Communications, 2000.