



UNIVERSIDAD DE BELGRANO

Las tesinas de Belgrano

**Facultad de Tecnología Informática
Ingeniería en Informática**

Simulación en Arena de un problema de colas en un
aeropuerto.

Nº 282

Martín Eurnekian

Tutor: Víctor M. Rodríguez Carro

Departamento de Investigaciones
Noviembre 2009

Índice

1.	Introducción	5
1.1.	Objetivos	6
1.2.	Alcances	6
1.3.	Límites	6
2.	Simulación	6
2.1.	Introducción	6
2.1.1.	Simulación y modelos	8
2.1.2.	Partes de un modelo	8
2.1.3.	Validación y verificación del Modelo	9
2.2.	Otros tipos de simulación	10
2.3.	Ejecución de una simulación	10
2.4.	Análisis de salidas de una simulación	12
2.5.	Ventajas y desventajas de la simulación	13
2.6.	Software de simulación	13
3.	Arena	14
3.1.	Introducción	14
3.2.	Modelos en Arena	16
3.2.1.	Procesos	16
3.2.2.	Datos	17
3.2.3.	Animación	17
3.3.	Ejecución de una simulación	17
3.4.	Reportes de una simulación	19
3.5.	Análisis de entradas con Arena	22
3.6.	Basic process panel	23
3.7.	Advanced process panel	23
3.8.	Otros paneles avanzados	23
3.9.	Comparación de escenarios en Arena	25
3.10.	Búsqueda de un escenario óptimo	25
4.	Simulación de colas en un aeropuerto	25
4.1.	La utilización de la Simulación en el análisis de la capacidad de los aeropuertos	26
4.2.	Descripción del problema	26
4.2.1.	El aeropuerto	26
4.2.2.	Los procesos del subsistema de partidas	29
4.3.	Modelo del Aeropuerto de Carrasco	30
4.3.1.	Objetivos y métricas del modelo	30
4.3.2.	Límites específicos del modelo	31
4.3.3.	Modelo en Arena	32
4.3.4.	Animación	41
4.4.	Simulación	42
4.5.	Resultados	44
5.	Conclusiones	46
6.	Líneas de investigación futuras	47
7.	Glosario	47
8.	Anexo A: Módulos principales de Arena	48
8.1.	El panel de Procesos Básicos	48
8.1.1.	Los módulos de flujo	48
8.1.2.	Módulos de Datos	52
8.2.	Panel de Procesos Avanzados	54
8.2.1.	Los módulos de flujo	54
8.2.2.	Módulos de Datos	62
9.	Anexo B: Diagrama de flujos del modelo	64
10.	Anexo C: Reporte de Arena	64
11.	Bibliografía	64

1. Introducción

La aviación comercial mundial ha tenido, desde sus comienzos a mediados del siglo XX, un crecimiento como pocas industrias han visto. Desde pequeños aeródromos donde sólo los más pudientes y osados viajaban en aviones con tecnología rudimentaria, hasta aeropuertos de hoy, como el de Chicago con 88 millones de pasajeros al año. El avance de esta industria tanto en lo tecnológico como en lo comercial ha llevado a todos sus participantes a desarrollar sus campos de acción al máximo, profesionalizando cada puesto de trabajo y agregando valor y procesos de análisis a cada uno de los sistemas intervinientes en las operaciones. Podemos destacar entre estos sistemas o subsistemas a las aerolíneas, donde probablemente encontremos el nivel más alto de sofisticación, con operaciones altamente complejas de coordinación y planificación de recursos que incluyen aviones, personal calificado de distintas especialidades, recursos de mantenimiento, rutas, etc. Existen hoy en día compañías como Delta Airlines, que vuela con más de 500 aviones unos 5000 vuelos diarios y transporta a más de 100 millones de pasajeros al año. Otros subsistemas incluyen el control de tráfico aéreo (ATC, por sus siglas en inglés) y los aeropuertos.

Todos estos ejemplos muestran claramente el tamaño y la complejidad existente en los actores de esta industria, así como enumeran una serie de sistemas que pueden estudiarse por separado, pero que tienen relaciones muy estrechas, hasta simbióticas. Este estudio se centrará en los aeropuertos que a pesar de ser un elemento de naturaleza estática, respecto al dinamismo de las aerolíneas, su complejidad es hoy en día altísima y se ha transformado en un elemento clave de la industria aerocomercial.

Básicamente, un aeropuerto es un conjunto de recursos que sirven al desenvolvimiento de las operaciones aeronáuticas. Contiene de grandes sistemas, como el vehicular, que consta de todos los accesos, áreas de estacionamiento y circulación de vehículos, el de Terminal de pasajeros, que consta de todas las áreas públicas y restringidas por donde pasan los pasajeros tanto en zonas de check-in, sectores de control de seguridad, áreas de pre-embarque, etc. y por último, el sistema del lado aire¹, que cuenta con pistas, calles de rodaje para aeronaves, plataforma de estacionamiento de aeronaves y sectores de mantenimiento. Cada uno de estas sistemas y sus respectivos subsistemas son esenciales para la operación del aeropuerto y cada uno presenta situaciones y problemáticas particulares. A su vez, todos son de una forma u otra el resultado de las enormes obras de infraestructura que implican los aeropuertos.

En nuestro caso vamos a centrarnos en el subsistema de la Terminal de pasajeros que tiene que ver con el movimiento de pasajeros de partidas, desde su arribo al aeropuerto, hasta que llegan a la puerta del avión.

El crecimiento vertiginoso de la industria, que según el Airports Council Internacional (ACI), seguirá siendo sostenido para los próximos 15 años, obliga a las instituciones responsables del manejo de aeropuertos estudiar cuidadosamente la demanda actual y sus proyecciones para mantener la infraestructura operativa con niveles de servicio aceptables. Esto se logra a través de inversiones en la ampliación de la misma para adecuarla a la proyección de la demanda. Estas inversiones son millonarias y por tanto mantener una relación óptima entre niveles de servicio mínimos e infraestructura ociosa (o sobre-inversión) es prioritario.

Para asegurar que el servicio brindado a los pasajeros en los aeropuertos tenga un confort establecido, la International Air Transport Association (IATA), ha desarrollado recomendaciones y normativas para diseñar aeropuertos y métricas para determinar los niveles de servicio brindados en un determinado momento. Para el caso del subsistema que vamos a analizar, IATA le otorga categorías a los aeropuertos según el nivel de confort con el que cuentan los pasajeros en todos los procesos de la operación de partida y esto tiene que ver con el espacio por pasajero que tienen las áreas que se utilizan. De esta manera un aeropuerto puede estar entre las categorías A y E según su nivel de confort. Por ejemplo, se considera que un aeropuerto debe tener, en su sector destinado a los chequeos de seguridad de los pasajeros, una superficie mínima de 1,4 metros cuadrados por pasajero para ser categoría A. O sea, que en el momento de mayor afluencia de pasajeros o cuando las colas son más largas, si el aeropuerto pretende mantener la categoría, el dimensionamiento de la sala destinada al chequeo de seguridad debe prever 1,4 metros cuadrados por pasajero. Esta medida puede darse por una serie de razones, por la afluencia de pasajeros, el tamaño de la sala y por la cantidad de recursos destinados al procesamiento de la seguridad de los pasajeros, así como el tiempo requerido para el proceso. El cambio de cualquiera de esas variables genera un nuevo escenario en la medición de categoría para ese subsistema.

En este contexto, los encargados del diseño de aeropuertos deben estudiar todos los subsistemas de los aeropuertos y su comportamiento relacionado con la demanda proyectada para poder tomar decisiones de diseño sobre las distintas áreas para asegurar el mantenimiento de la categoría IATA definida. Estos estudios se denominan análisis de capacidad de aeropuertos.

1. Se refiere a todo el área de un aeropuerto destinada a pasajeros en la jerga aeronáutica.

1.1. Objetivos

Los objetivos de esta tesina son:

- Demostrar que la simulación es un método que ha probado ser efectivo en la realización de análisis de capacidades en aeropuertos.
- Presentar y estudiar la herramienta de simulación Arena, y mostrar que es aplicable a la simulación de aeropuertos.
- Utilizar la herramienta Arena para simular el sistema de partidas de la nueva Terminal del aeropuerto de Carrasco en Montevideo, Uruguay, y determinar en qué categoría IATA se encuadra.

1.2. Alcances

El presente trabajo pretende realizar una breve introducción a la simulación para el lector que tiene conocimientos del área de sistemas y computación. Sumado a esto se realizará una presentación de la herramienta Arena que comprenderá su concepto general y en particular de sus funciones y módulos principales; así como una breve descripción de cómo utilizar el sistema para realizar una simulación.

A partir de la introducción de las disciplinas generales de simulación y de la herramienta Arena, se construirá con ella un modelo del sistema de partidas de la Terminal de pasajeros del aeropuerto de Carrasco, para determinar su categoría IATA. Para esto, primero se hará una descripción pormenorizada del sistema y su marco teórico, incluyendo la normativa IATA que regula la categorización de aeropuertos. Para luego plantear el problema en términos implementables en Arena, y pasar a su desarrollo. Luego de esto, y contando con las salidas de la simulación, se explicarán las conclusiones para mostrar la categoría en la que se clasifica el aeropuerto en estudio.

1.3. Límites

El trabajo se limitará a presentar la herramienta Arena a efectos de que el lector entienda su propósito, funcionamiento general y su forma de uso. No se pretende hacer un juicio de valor sobre la calidad de la herramienta ni una comparación con otros productos del mercado. Se darán a conocer sus funciones principales y básicas, pero solo se ahondará en las más avanzadas que sean necesarias para la simulación del aeropuerto que se realizará en la sección 3 del trabajo y esto se hará en el marco de esa simulación. El resto de las funciones avanzadas se mencionarán para que el lector conozca su existencia y pueda investigar al respecto.

La simulación del sistema de partidas de la Terminal de pasajeros del aeropuerto de Carrasco incluirá el arribo de pasajeros a la Terminal, el proceso de check-in, el proceso de chequeo de seguridad y el proceso de migraciones. No incluye la simulación de las etapas posteriores al proceso de migraciones. Es decir, no se modelará el preembarque ni el paso de los pasajeros de la puerta del aeropuerto al avión. En el caso de los recursos destinados a los procesos de seguridad y migraciones, se tomarán medidas estándar definidas en las recomendaciones IATA, sin tener en cuenta las variaciones que cada compañía aérea o la Dirección de Migraciones puedan realizar sobre la cantidad y los horarios de los recursos. También, para simplificar el modelo, se asume que las operaciones de despegue de los aviones se hacen a los horarios definidos, sin demoras particulares, ni generales debido a meteorología y otras circunstancias.

2. Simulación

2.1. Introducción

La simulación, en el contexto de la informática, es la imitación de un sistema real a través de la utilización de un modelo lógico construido en una computadora. Un modelo es una descripción abstracta del sistema a imitar, que a través de relaciones lógicas y matemáticas lo representa. El mismo puede ser interpretado por una computadora permitiendo realizar la simulación.

La simulación es una colección de métodos muy potente que permite ayudar a diseñar y evaluar sistemas del mundo real, tanto existentes como teóricos. Esto nos permite imitar procesos industriales, servicios, procesos comerciales, procesos académicos y de diversa índole. Algunos ejemplos incluyen fabricas, con todos los procesos, materiales y actores intervinientes; servicios como bancos u hospitales; y procesos comerciales como un almacén de distribución o un restaurante de comida rápida.

La realización de una simulación supone la disponibilidad de un modelo que puede ser el producto del análisis de un sistema existente o el diseño de uno a construir. En ambos casos el objetivo de la simulación es el de conocer cómo se comporta el mismo, ya sea para compararlo con otro sistema o con otra configuración del mismo sistema.

Según Kelton et al (2007)², en un paso previo a la simulación, la construcción del modelo puede ser en sí misma un proceso enriquecedor, ya que nos vemos obligados a entender el sistema a estudiar y a formalizar y documentar su funcionamiento. A partir de este proceso se suelen generar alternativas para su configuración y optimización.

Para entender la utilidad de la simulación y darle una clasificación dentro de otras técnicas usadas para observar y medir la performance de sistemas reales, Law (2007)³ describe con el cuadro de la figura 1 las distintas formas de estudiar un sistema.

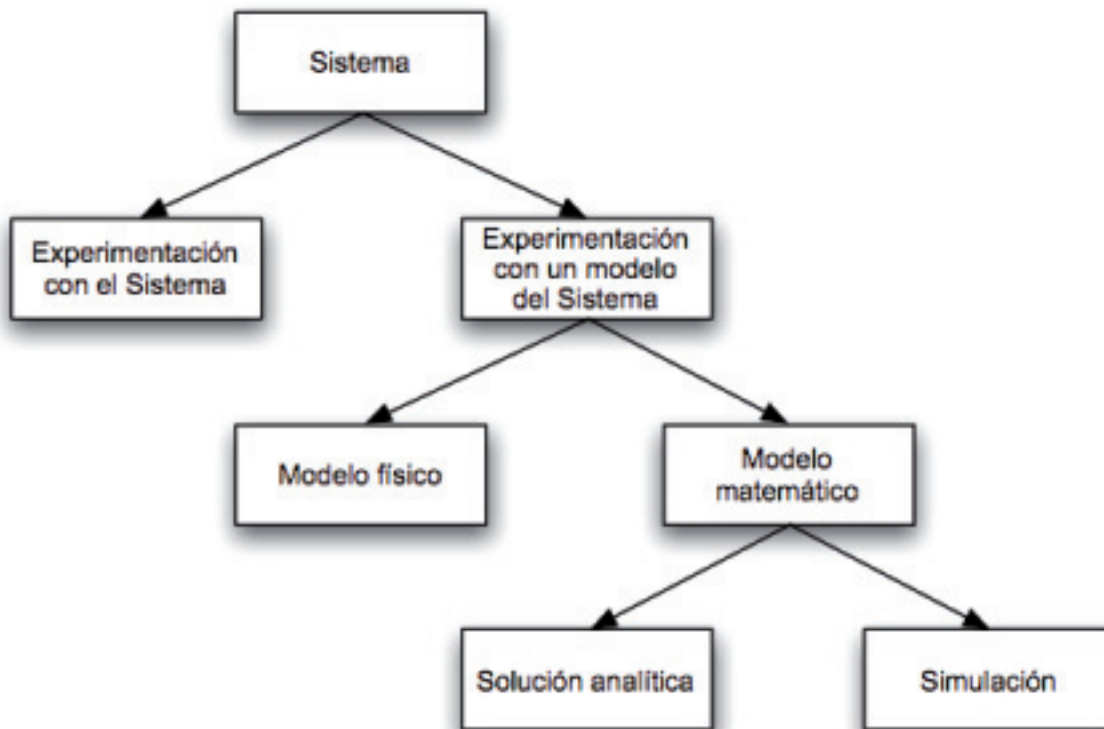


Figura 1 Formas de estudiar un sistema según Law (2007)

La elección de la simulación como método de estudio del sistema tiene que ver con la inviabilidad de la realización de los otros métodos propuestos. Es decir, no parece tener sentido generar un modelo de un sistema para luego simular su funcionamiento si es posible manipular y estudiar el sistema real. O sea que, tomando el caso de un proceso de check-in de un aeropuerto, podríamos querer estudiar el comportamiento de las colas que se forman si agregamos 10 posiciones de atención a pasajeros. En este caso, para experimentar con el sistema real, habría que realizar una obra civil, fabricar muebles, extender los sistemas de manejo de equipajes y tomar personal. Todo esto es muy costoso y hace demasiado oneroso el esfuerzo para realizar una prueba. Por esta y otras razones de complejidad que hacen imposible experimentar con el sistema real se recurre a los modelos. Estos suelen ser, en el marco de este trabajo y en el de la simulación en general, modelos lógicos o matemáticos usados por una computadora para llevar a cabo la simulación. También existen modelos físicos de sistemas del mundo real, estos son prácticos en algunas disciplinas, pero generalmente muy costosos. Un ejemplo de estos últimos pueden ser modelos a escala de aviones o autos, que son sometidos a pruebas de túnel de viento para estudiar su aerodinámica, o en un caso más simple, una maqueta de un edificio es un modelo físico. Para el caso de sistemas simples, es posible conseguir buenos resultados mediante técnicas que generan un modelo matemático que da un resultado directo del funcionamiento del sistema. En sistemas complejos del mundo real es muy difícil generar estos modelos ya que la complejidad los torna inmanejables. Por todas estas razones, es que tanto Kelton et al (2007) como Law (2007) dejan claro que la simulación es una técnica muy útil para una variedad importante de problemas del mundo real y que esto es posible gracias a un versátil manejo de la complejidad.

2. Kelton, D., Sadowsky, R., Sturrock, D.: "Simulation with Arena", 4ta. edición, Mc Graw Hill (2007)

3. Law, A.: "Simulation modeling and analysis", 4ta. edición, Mc Graw Hill (2007)

2.1.1. Simulación y modelos

La simulación de sistemas del mundo real requiere, primero, de un modelo sobre el cual se realizará el análisis. En segundo lugar, es necesario, tener claro un aspecto del mismo a estudiar. Idealmente, tendremos una pregunta específica para responder o una variable o proceso a mejorar. Ejemplos de esto son, cuántas posiciones de check-in se necesitan en un aeropuerto para que las colas nunca superen las 10 personas, o qué nivel de utilización tiene una máquina de rayos X de un puesto de control de seguridad. La mayoría de las veces, los modelos que se construyen, tienen uno o más componentes de naturaleza aleatoria. Tomando el caso del aeropuerto o el banco, nosotros podemos prever horarios picos de llegada de pasajeros o clientes, pero nunca sabemos a qué hora exacta va a llegar cada cliente. Es por eso que en el modelo se utiliza una distribución probabilística para generar aleatoriamente el arribo de clientes. Los modelos que tienen uno o más componentes aleatorios (la mayoría, y sobre los cuales nos focalizaremos) se denominan estocásticos. En contraste, los modelos que tienen una salida determinada netamente por la entrada se denominan determinísticos.

La naturaleza aleatoria de un modelo genera que su salida o resultado también sea aleatorio. Eso es un aspecto clave a considerar cuando se realiza una simulación y sobre todo a la hora de analizar los resultados. Thesen y Travis (1991)⁴ llaman a este principio RIRO (random input random output).

Para poder entender el espíritu del funcionamiento de la simulación, debemos primero entender la naturaleza de diversos aspectos clave del sistema a estudiar. Sin duda, el más importante, y el que transforma al objeto de estudio en un sistema plausible de ser analizado vía simulación es el de la naturaleza temporal del sistema. Es decir, si los cambios que se producen en los estados del sistema son graduales en el tiempo haciendo necesario el análisis de este como una medida del conjunto de números reales, probablemente nos encontremos ante un modelo continuo. Para entender mejor esta clasificación de modelos, Thesen y Travis (1991) comparan: "... en contraste con los continuos, los modelos discretos describen sistemas en los que se asume que los cambios de estado se dan instantáneamente en respuesta a cierto evento u ocurrencia súbita". Por ejemplo, un modelo continuo puede ser el de un avión moviéndose en el espacio, para cada valor de t (tiempo) la posición va ser distinta; mientras que el sistema de atención a clientes de una sucursal de banco representa claramente un modelo discreto, no vamos a encontrar una cantidad de clientes (c) distinta para cada valor de t , c solo cambia cuando entra o sale un cliente del banco (evento). La simulación por computadora a la que este documento se refiere es la que trabaja con modelos discretos. Según Thesen y Travis (1991), cuando elegimos modelar un sistema del mundo real usando simulación con eventos discretos (discrete-event) renunciamos a la posibilidad de capturar un grado de detalle solo describable como un cambio suave y continuo. Pero así conseguimos una simplicidad que nos permite capturar las características importantes de muchos sistemas que son demasiado complejos para ser modelados usando modelos continuos.

En contraposición con la simulación orientada a eventos, como lo vimos con los modelos discretos, encontramos la simulación orientada a procesos. En la primera tanto el modelo como el trabajo computacional están centrados en los eventos que ocurren en el sistema. De esta manera se pueden realizar modelos bastante simples que tienen la suficiente versatilidad para ser implementados en computadoras. Sin embargo, cuando el modelo es muy complejo la implementación en esta orientación puede ser muy tediosa. Es por eso que existe la orientación a procesos en la simulación. Tomando como base a las entidades, esta orientación describe todos los procesos que cada una realiza durante su paso por el sistema, simplificando los modelos de sistemas complejos. Algunos lenguajes de simulación como el SIMULA, se codifican con orientación a procesos, pero internamente hace el trabajo computacional en orientación a eventos.

2.1.2. Partes de un modelo

Para explicar como funciona una simulación vamos a comenzar por conocer las partes que forman un modelo:

Entidades: Son la parte dinámica del modelo. Las entidades son las que afectan al estado del sistema y a otras entidades. Usualmente se crean, atraviesan el sistema y luego son desechadas. Pueden existir varios tipos de entidades en un modelo y normalmente representan a objetos reales en el sistema. Además, de cada tipo pueden existir múltiples instancias conviviendo en el modelo durante la simulación.

Atributos: Son una característica común de las entidades, es decir, se comparte entre todas las entidades de un tipo, pero puede individualizar a cada instancia.

Variables: Las variables, siempre globales, permiten definir una característica general del sistema. Las variables son al sistema lo que los atributos a una entidad. Pueden ser accedidas y modificadas por cualquier entidad. El reloj del sistema, encargado de llevar el tiempo simulado (distinto del tiempo que lleva realizar la simulación) es una variable global que está presente en toda simulación.

4. Thesen, A., Travis, L.: "Introduction to simulation", Proceedings of the 1991 Winter Simulation Conference

Recursos: Son quienes dan servicio a las entidades para procesos u otros trabajos. Generalmente en los sistemas del mundo real son máquinas o personas que trabajan sobre las entidades. La cantidad de recursos disponible en el sistema durante la simulación puede variar debido a que se representan casos como turnos para personal, etc.

Colas: Cuando una entidad necesita ocupar un recurso que no está disponible (por estar dando servicio a otra entidad) debe esperar a su turno para acceder al recurso. En ese caso la entidad espera en una cola. El comportamiento de la cola se puede manejar a discreción por parte de quien diseña la simulación.

Acumuladores estadísticos: Para que la salida de una simulación contenga información útil para quien realiza el experimento se necesita acumular datos durante la el proceso. Para esto se crean los acumuladores que llevan contabilización de distintos parámetros del modelo para que luego la información necesaria este disponible al terminar el experimento.

Eventos: Un evento es lo que pasa a un determinado instante de tiempo simulado que genera o dispara una serie de cambios en el sistema. En la simulación de eventos discretos (discrete-event) solo se generan cambios en los estados del sistema, nunca entre eventos.

Reloj de simulación: Es la variable más importante. Contiene el tiempo actual simulado. Considera el valor del último evento ejecutado en lugar de correr tomado valores continuos como un reloj normal.

Es muy importante definir como parte de cualquier simulación definir las condiciones de comienzo y fin de la misma. Puede ser procesar un cantidad fija de entidades o recorrer un tiempo (simulado) establecido, o funcionar hasta que se cumpla una condición.

2.1.3. Validación y verificación del Modelo

Para que una simulación sea exitosa y se obtengan los resultados necesarios para tomar decisiones o conocer a fondo el sistema a analizar, es muy importante prestar suma atención al nivel de detalle del modelo generado.

En primera medida, un modelo es siempre una simplificación abstracta del sistema real. Lo esencial es que dicha simplificación sea la adecuada para reproducir el comportamiento del sistema original en el modelo dejando de lado los detalles que no son esenciales o que no proporcionan valor al aspecto a estudiar. Si generamos un modelo muy simplificado, vamos a perder comportamientos del sistema que seguramente son importantes para conseguir un resultado real, parecido al resultado producido por el sistema estudiado. En cambio, si nos obsesionamos con reflejar el sistema exacto, tal cual es, seguramente perdamos tiempo modelando partes que no agregan valor al resultado de sistema y que seguramente harán al modelo más difícil de leer y propenso a errores. Para llegar a un modelo con el grado necesario de detalle, Thesen y Travis (1991) recomiendan comenzar por modelar el sistema más simple posible para luego iterar el proceso de modelado agregando en cada iteración complejidad al modelo, hasta llegar a uno que haya incorporado todo el comportamiento necesario.

Una vez terminado el modelo, el siguiente paso a realizar es verificarlo. Este proceso tiene como objetivo comprobar que se comporta tal cual fue concebido. Más allá de asegurarse que el modelo refleje el sistema del mundo real, el proceso de verificación tiene que ver con la comparación del modelo con su especificación realizada a partir del estudio del sistema real. A simple vista, esto parece una tarea simple, pero en modelos complejos donde se producen varios procesos de manera simultanea, generalmente existen interacciones y resultados que están fuera de toda previsión y no siempre son fáciles de visualizar. Es por eso que la gran mayoría de los autores que escriben sobre el tema recomiendan generar diagramas de flujo del modelo y animaciones de la simulación para poder ver el sistema rápidamente y comenzar con una verificación conceptual del modelo a simple vista. A partir de esto se recomienda generar pruebas sobre el modelo para ver sus resultados y chequear si estos son coherentes respecto a lo esperado. Así como en la fase de testeo de un sistema de software tradicional, se realiza una batería de pruebas sobre el sistema para determinar la corrección del mismo y quitar todo comportamiento o código indeseado (debugging), en simulación la fase de verificación tiene un contenido similar. Para llevar a cabo esta tarea se deben realizar pruebas al modelo en distintas condiciones, normales y anormales. Dentro de lo que llamamos condiciones anormales de prueba están todos los stress-tests que llevan los parámetros del sistema a extremos críticos, altos y bajos para ver como reacciona el sistema y detectar errores que en condiciones normales pasan desapercibidos.

Varios autores coinciden en que luego de las pruebas se debe reunir al equipo de simulación con los expertos en el sistema a estudiar y los interesados en el resultado de la simulación para realizar sesiones de verificación del modelo. Sea por revisiones del funcionamiento de la animación como por el uso de la técnica conocida como *structured walk-through* (ver Law (2007) p.249 y p.245). En ambos casos, la idea es que con la inclusión de expertos en el tema y demás interesados se encuentren comportamientos válidos e inválidos de la simulación y también se revise la salida para ver si puede considerarse coherente

por parte del equipo. Kelton et al (2007) recomiendan también buscar propuestas de cambio en el equipo de verificación. Así como reconoce que es casi imposible verificar completamente un modelo complejo y que no existe una única y correcta técnica para realizar la tarea. Lo importante es que tanto el equipo de desarrollo como los usuarios del sistema se sientan conformes con el comportamiento del modelo.

La verificación es una parte del desarrollo del modelo que nos permite perfeccionarlo y llegar a un producto que funciona así como fue definido. Una vez obtenido ese producto se debe pasar a una tarea aún más compleja, la de validación del modelo. Según Kelton et al (2007) la tarea de validación es la de asegurarse que el modelo se comporta igual que el sistema del mundo real. Para ello Law (2007) propone realizar lo que llama validación de resultado (result validation). Esto consiste en comparar los resultados del sistema real con los de la simulación. Esta tarea es extremadamente compleja ya que no siempre existe un sistema real para comparar, y muchas veces si existe no hay datos históricos o estos no son confiables. En el caso de que existan datos históricos suficientemente buenos como para usar en una comparación directa, se validaría el sistema si los resultados de la simulación son lo suficientemente cercanos a los datos recabados como para que el equipo de trabajo encuentre coherencia en los mismos. Law(2007) propone la realización del llamado Test de Turing en el cual se presentan los datos históricos junto con los resultados de la simulación a un equipo de expertos en el sistema real sin que ellos sepan cual es cual. Luego se les pide que identifiquen a qué sistema pertenece cada set de resultados. Si no lo logran se considera que el sistema esta validado. Por otra parte, si no existiera la posibilidad de encontrar datos históricos usables, es posible que la realización de la validación sea imposible. Para lo cual Law(2007) propone generar otra serie de medidas para llegar a una conclusión sin datos. Estas son la comparación del sistema con la opinión de expertos en la materia, para la cual se considera que si ellos están de acuerdo con los datos extraídos de la simulación el modelo tiene una validación facial (face validity); la comparación con otros modelos similares que existan, lo cual es una tarea peligrosa que puede llevar a conclusiones erróneas acarreado errores de un modelo a otro; y por último usar la animación para validar con el equipo de trabajo el modelo. Resumiendo, y citando a Kelton et al (2007), si no existe la posibilidad de acceder a datos del sistema real para comparar con el modelo, hay que concentrar todos los esfuerzos en la verificación.

2.2. Otros tipos de simulación

Además de las clasificaciones generales que aparecieron anteriormente en el texto, existen otras en las que se pueden enmarcar los modelos y los experimentos de simulación según sus características.

La mayoría de las simulaciones pueden ser clasificadas como acotadas en el tiempo, *terminating*, o en régimen permanente, *steady-state*. Según Kelton et al (2007) las *terminating* son aquellas "...en las que el modelo dicta condiciones específicas de comienzo y fin como reflejo natural de cómo el sistema objetivo opera en realidad". Es decir, la simulación siempre termina por una condición inherente al modelo. Por ejemplo, un banco que abre de 10 a 15 hs., el modelo empieza vacío a las 10 hs., a partir de ahí llegan los clientes que son atendidos durante el día hasta las 15 hs. cuando el último cliente se retira del banco.

En el caso de las simulaciones *steady-state* se supone que el sistema real opera en tramos largos, teóricamente infinitos. En estos casos las condiciones iniciales de la simulación no son importantes, ya que en el sistema real no existen (porque siempre está funcionando), y por lo tanto no distorsionan los resultados de la misma.

En secciones anteriores hemos hecho referencia a los modelos discretos y continuos, existen casos en que sistemas no pueden ser clasificados como uno u otro, los mismos se denominan combinados discreto-continuo. Estos casos son raros y más difíciles de implementar.

La importancia del tiempo simulado para esta disciplina es enorme, eso es porque siempre nos enfrentamos con modelos de los llamados dinámicos, donde es importante el tiempo simulado. En cambio existen los modelos estáticos, en los que el tiempo simulado no juega un papel importante y la simulación avanza y termina sin nunca hacer referencia al mismo.

2.3. Ejecución de una simulación

Una simulación es en muchas maneras como un sistema informático cualquiera. Por eso para realizar con éxito el desarrollo de un modelo y de una simulación se deben realizar pasos similares a los que se realizan en el desarrollo de un sistema informático. Hay mucha bibliografía respecto a dichos pasos, pero no siempre se encuentra un método aceptado para seguir en una simulación. De la misma manera que en la computación en general existen distintas visiones y métodos para realizar análisis y desarrollo de sistemas. De hecho muchas técnicas propuestas tienen cosas en común con aquellas del área de ingeniería de software. En este caso tomaremos como guía lo propuesto por Law (2007). Por ser el más estructurado y porque se trata de un especialista muy reconocido en la practica de simulación.

Más adelante, en la figura 2 vemos el diagrama estructurado que propone Law (2007), con los pasos a seguir en un estudio de simulación, a continuación explicaremos la aplicación de cada paso.

Formular el problema y planear el estudio.

Generalmente los estudios de simulación se solicitan para resolver problemas de diseño o funcionamiento de algún sistema del mundo real. Quien encarga la simulación generalmente tiene una idea de cual es el problema y puede dar un soporte escrito con el problema y lo que espera cómo solución. En la mayoría de los casos, a partir de una investigación, se descubre que esta descripción del problema es incorrecta o está mal expresada. Por lo cual debemos rehacerla en un proceso iterativo de recopilación de información y consultas con los interesados. Luego se deben formalizar los objetivos del estudio, los alcances del mismo, el tiempo disponible y la plataforma o herramienta que se usará para este. En el caso de este trabajo, Arena.

Recopilar datos y definir el modelo.

En esta etapa se debe juntar toda la información disponible, tanto del sistema a modelar como de sistemas similares existentes (para el proceso de validación). Al mismo tiempo se debe comenzar a definir el modelo a partir de los datos recopilados y documentar el proceso. Law (2007) propone un documento de hipótesis (Assumptions Document), mientras que Kelton et al (2007) lo llama Documento de Especificaciones usando la terminología habitual de ingeniería de software. En esta etapa se debe decidir sobre el nivel de detalle con el que se construirá el modelo ya que debemos comenzar a definir el mismo. Este proceso también se puede hacer en forma iterativa comenzando con un modelo simple y luego incrementar su complejidad en sucesivas iteraciones.

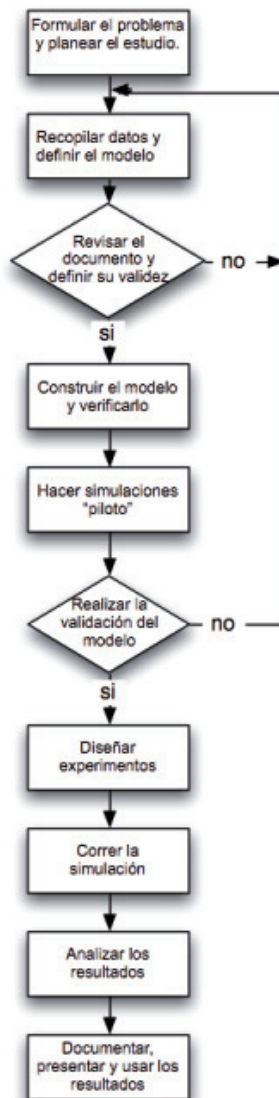


Figura 2 Pasos de un estudio de simulación. Law (2007)

Revisar el documento y definir su validez.

Para tomar esta decisión se propone reunir al equipo de trabajo y los interesados para revisar el documento y realizar un *structured walk-through*. Es importante comenzar a desarrollar el modelo teniendo la mejor documentación posible, ya que nos acortará el tiempo y nos ahorrará volver a revisar el documento con mayor frecuencia. En caso de decidir que el documento no es válido volveremos a al paso 2 para mejorarlo.

Construir el modelo y verificarlo.

En este paso tomaremos la plataforma elegida para implementar el modelo y lo haremos realidad. A partir de eso haremos la verificación del mismo como la describimos en la sección 2.1.3 del presente trabajo.

Hacer simulaciones "piloto".

A partir del modelo implementado se harán simulaciones para obtener los resultados del mismo y proceder a la validación del modelo.

Realizar la validación del modelo.

Tal cual descrito en la sección 2.1.3 debemos validar el modelo.

Diseñar experimentos.

En esta etapa se deben tener en cuenta las preguntas que debemos responder para lograr el objetivo de la simulación y diseñar experimentos a realizar con el modelo. Esto significa que se deben elegir distintos sets de parámetros a ingresar en el modelo así como la cantidad de simulaciones etc.

Correr la simulación.

Aplicar los experimentos diseñados en el paso anterior para obtener los resultados de la simulación.

Analizar los resultados.

Mediante herramientas estadísticas se analizan los resultados para responder a las preguntas objetivo, ya sea la elección de un diseño de sistema o la maximización de un aspecto del mismo.

Documentar, presentar y usar los resultados.

La documentación del estudio es muy importante. No solo de los resultados sino del modelo y de la documentación intermedia del proyecto. Ya sea para futura consulta o para respaldo de los resultados. Este proceso debe mantenerse durante todo el estudio de simulación.

La presentación y uso de los resultados de un estudio de simulación exitoso se usa, generalmente, para tomar decisiones sobre el sistema estudiado y por tanto debe tenerse especial cuidado en su presentación y uso.

Respecto a este último punto, Kelton et al (2007) recalca que no siempre el desarrollo de un buen modelo de simulación significa haber tenido éxito en trabajo y que muchas veces se puede estar trabajando sobre objetivos equivocados. Por lo cual, a pesar de haber conseguido el objetivo, el trabajo es un fracaso. Esto ocurre cuando el objetivo fijado no es el esperado por quien ordena el trabajo y por eso recomienda que siempre se discuta el objetivo claramente para quitar ambigüedades.

2.4. Análisis de salidas de una simulación

Muchas veces se confunde el objetivo de la construcción de un modelo para simulación. Es común creer que el esfuerzo mayor en un estudio de simulación es la construcción del modelo en sí mismo, y que una vez construido solo hay que correrlo para obtener la respuesta deseada. Law (2007) define a la simulación como "...un experimento de muestreo estadístico basado en computadora." y por el principio RIRO de Thesen y Travis (1991) sabemos que debido a la naturaleza aleatoria de las entradas de una simulación las salidas tienen un alto componente aleatorio. Por todas estas razones es que se debe tener mucho cuidado al interpretar las salidas de una simulación. Nunca deben tomarse los resultados absolutos de una corrida del experimento de simulación como la respuesta. La única forma de interpretar los resultados de una simulación es a través de inferencia estadística. Y para ello debemos acumular una cantidad de datos significativa realizando varias corridas del experimento.

Es importante destacar que tampoco pueden ser aplicadas directamente las técnicas tradicionales de inferencia estadística basadas en datos independientes y distribuidos idénticamente (IID, por sus siglas en inglés, independent and identically distributed), a menos que el software de simulación utilizado prevea la posibilidad de crear replicaciones independientes. Para el primer caso, hay una gran cantidad de técnicas para poder interpretar los datos provenientes de los resultados de simulaciones, no es nuestra intención explicarlas en esta tesina, pero en Law (2007) (p.485) y en Nakayama (2006) se pueden encontrar las más usadas. Una de las razones por la cual no vamos a ahondar en estas técnicas es que Arena proporciona una respuesta con inferencia estadística ajustada automáticamente.

2.5. Ventajas y desventajas de la simulación

La simulación es, hoy en día, una herramienta que se ha consolidado en el ambiente de la informática, la investigación operativa y la ingeniería industrial y de sistemas como la más popular por su versatilidad y capacidad de análisis para modelos de gran complejidad. Sin embargo, como toda metodología de análisis, tiene sus desventajas y puntos débiles.

Sin duda el punto más destacado en la bibliografía especializada tiene que ver con la aleatoriedad encontrada en los resultados de los estudios debido a la misma condición de sus entradas. Esto trae gran variabilidad en los resultados y por ende los hace complejos de interpretar. Por otro lado, no solo los resultados son difíciles de obtener, sino también el modelo presenta complicaciones para su desarrollo, casi en la misma medida del diseño y desarrollo de un sistema de software.

Del lado de las ventajas, encontramos unánimemente la capacidad de la simulación de dar conocimiento y capacidad de manipulación sobre sistemas del mundo real que de otro modo sería imposible obtener.

La simulación es una herramienta muy útil para resolver una gran cantidad de problemas, pero no es apropiada para todos los problemas que se presentan. Es importante estudiar bien el problema y estar seguros de que la simulación es la herramienta adecuada para el mismo, ya que aplicarla es bastante complejo y si existiera otra metodología apta para la resolución del mismo, generalmente será la mejor opción.

2.6. Software de simulación

Siempre se considera que los estudios de simulación se realizarán en una computadora. A pesar de ser posible, a veces, su ejecución manual, la complejidad de los modelos y la velocidad actual de las computadoras hacen totalmente desaconsejable esta alternativa. Por esto el software de simulación juega un papel fundamental en esta disciplina.

Existen distintas formas de usar una computadora, y por ende software, para realizar simulaciones. Podemos utilizar programas específicamente diseñados para realizar estudios de simulación y modelos, o podemos utilizar lenguajes de programación de propósitos generales (como C o Java) para programar los modelos y ejecutar los estudios.

Los lenguajes de propósitos generales proveen al analista o programador gran flexibilidad para construir el modelo y existen librerías creadas para distintos lenguajes que traen implementadas funciones específicas para simulación. Sin embargo, debido a lo específico de la tarea y a la complejidad de los modelos, la elección de estos lenguajes para realizar esta tarea no es generalmente recomendable. La programación y el diseño de los programas puede ser muy compleja y gracias a las herramientas de desarrollo específicas para simulación la tarea se hace con mayor velocidad y simpleza. Estos generalmente cuentan con ambientes especiales para simulación con todas las funciones y módulos necesarios para realizar los modelos teniendo en cuenta su facilidad de mantenimiento y ayudando al programador a evitar los errores.

Dentro de los programas de simulación encontramos dos grandes áreas. La de los sistemas de simulación general, que prevén los módulos usuales de la simulación aplicada en general, y la de las herramientas orientadas a aplicaciones específicas. Estas últimas son creadas con un campo de aplicación en mente, una industria o un problema específico como ser el modelado de call centers, aeropuertos, etc.

Los sistemas de simulación general, como Arena, usan, para que el usuario genere los modelos, una aproximación al mismo desde el punto de vista de procesos. Es decir, que sin importar el manejo interno de la simulación discreta, el usuario genera el modelo construyendo diagramas de flujos con procesos y recursos que las entidades recorren mientras se produce la simulación. Además estos sistemas presentan una serie de características comunes y, según Law (2007), deseables. En primer lugar, un sistema debe ser flexible en sus construcciones y módulos de modo que permita representar una gran variedad de sistemas reales, y que si esto no fuese posible se pueda extender el sistema ya sea construyendo nuevos módulos a partir de los existentes o agregando rutinas de lenguajes de programación tradicional. Esto siempre debe permitirse sin olvidar amigabilidad para el usuario. También es importante que el sistema posea un poderoso módulo de testeo y corrección de errores que pudieran encontrarse en el modelo. Los resultados de una simulación son muy importantes, ya que determinan el éxito de un desarrollo. Estos deben ser de buena calidad así como configurables por el usuario para tener datos específicos o exportar en forma de archivo electrónico a otras aplicaciones. También es muy útil que el sistema permita la importación de datos de otras aplicaciones para el uso en un experimento. Como hemos visto anteriormente, la animación es una herramienta muy útil y utilizada en esta disciplina, por eso, es deseable que el sistema que utilizemos tenga un buen manejo de las mismas. Esto debe incluir la posibilidad de representar el sistema de manera muy simple acompañando el proceso de generación del modelo o de manera compleja realizando animaciones en tres dimensiones producidas con posterioridad a la ejecución

de la simulación. Otro aspecto clave es el de la generación de números aleatorios, esto es de suma importancia para el resultado de los experimentos y su validez estadística. Es por eso que debemos contar con un generador robusto de números aleatorios que permita la generación de al menos 100 corridas (streams) para los distintos usos del sistema, permitiendo a los resultados de varias ejecuciones ser IID, de manera que se puedan analizar con inferencia estadística estándar. Otra característica deseable es la presencia de un optimizador, esto permite encontrar valores cuasi óptimos para maximizar resultados a partir de una serie de parámetros de entrada de la simulación. Estas son algunas de las funciones que debemos buscar en un sistema de simulación. Como veremos en la próxima sección, Arena cumple y excede estas expectativas.

3. Arena

3.1. Introducción

En esta sección introduciremos una herramienta de simulación, Arena. Esta se ubica en lo que anteriormente definimos como sistema de simulación y permite, en un ambiente gráfico de Windows, modelar gran cantidad de sistemas y realizar simulaciones obteniendo resultados y analizando los mismos para facilitar la toma de decisiones.

Arena es un ambiente completo de desarrollo de simulaciones. Provee la capacidad de generar modelos, generar datos aleatorios como entradas, correr los modelos bajo una serie de condiciones parametrizables, generar animaciones, generar reportes estándares y personalizados, comparar distintos modelos, buscar optimización de modelos e integrar todo el proceso con las aplicaciones más usadas del mercado como Microsoft Office, bases de datos y lenguajes de programación. Kelton et al (2007) dice: "Arena combina la facilidad de uso de un sistema de desarrollo de alto nivel con la flexibilidad de los lenguajes de simulación y hasta con lenguajes de propósito general como Visual Basic o C...".

La facilidad de uso lograda por Arena tiene como característica principal su interfase gráfica que es intuitiva e integrada para todos los posibles usos de la aplicación. En la figura 3 vemos su pantalla principal, donde encontramos las secciones principales de la misma.

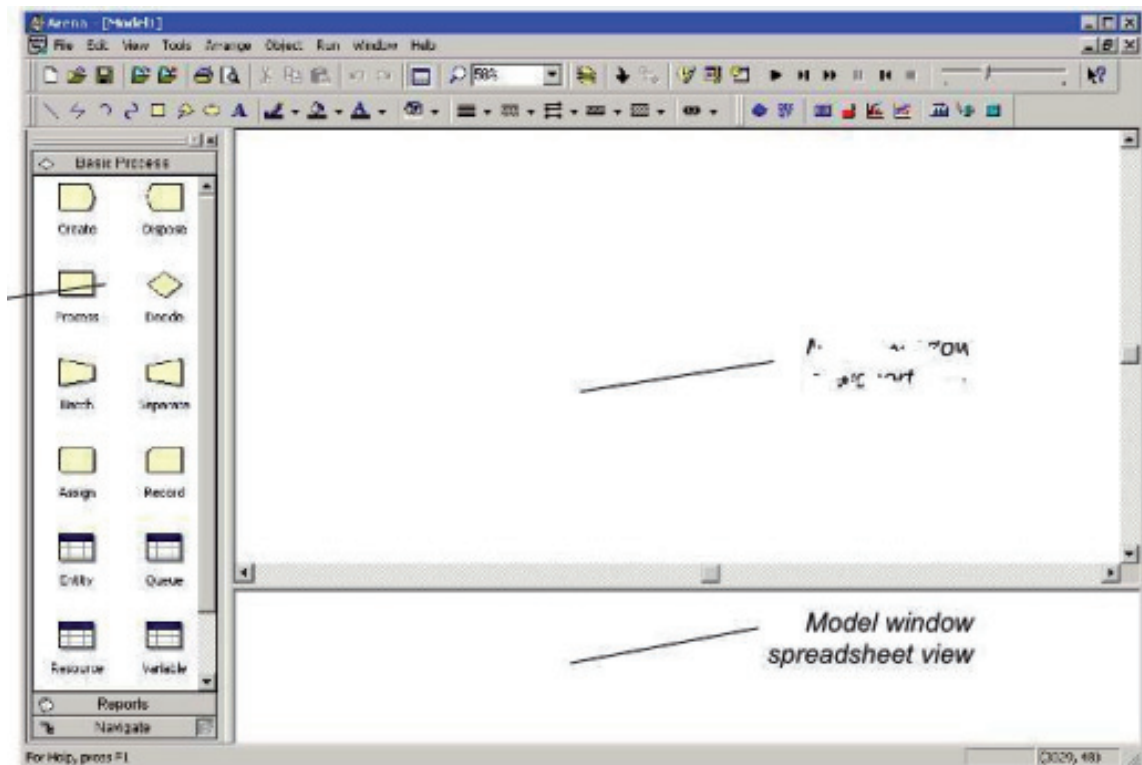


Figura 3 Pantalla principal de Arena

Sobre la derecha de la ventana encontramos las vistas generales del modelo. En esta área de trabajo se genera el modelo y la simulación. En la sección superior de la misma está la parte gráfica donde el diagrama de flujos del modelo y la animación se pueden ver y trabajar. En la inferior encontramos un área

de datos que nos permite editar las propiedades de cualquier módulo seleccionado. A la izquierda de la ventana encontramos la barra de proyecto, en ella encontramos distintas pestañas que controlan todos los aspectos de una simulación. Los módulos que representan los distintos procesos en un modelo se encuentran ordenados por clase dentro de paneles representados en las pestañas. Desde allí el usuario los toma para agregarlos a un modelo en desarrollo. Además hay otras dos pestañas que nos permiten ver otros aspectos de la simulación, la primera, llamada Reports muestra el árbol que representa la estructura de los reportes con los resultados de los experimentos, y la segunda, Navigate nos permite hacer paneos por áreas de trabajo grandes en modelos complejos. Ahí podemos generar vistas personalizadas que nos llevan directamente a áreas predefinidas del modelo lógico o la animación.

Arena funciona construido sobre el lenguaje de programación para simulaciones SIMAN. Esto proporciona una combinación de facilidad de uso por su interfaz y la versatilidad de un lenguaje de programación, pero esta combinación se explota de manera más interesante para el usuario gracias a su estructura jerárquica. Esta estructura permite armar construcciones de nivel bajo y combinarlas para formar una complejidad y funcionalidad mayor que se inserta en una estructura de un nivel superior en la jerarquía, dando facilidad de uso y mayor nivel de abstracción. Esto permite al usuario básico usar construcciones de alto nivel sin la necesidad de conocer como funciona internamente y al usuario experimentado generar nuevos comportamientos y funcionalidades. De la misma manera el sistema trae incluidas construcciones de casi todos los niveles para dar una muy alta flexibilidad a los modelos sin la necesidad de codificar para generarlos.

En la figura 4 vemos un esquema que muestra los distintos niveles de la estructura jerárquica de Arena. En el nivel más alto se encuentran los templates, estas son las construcciones de mayor complejidad que pueden ser creadas por los usuarios o ser incluidas en Arena. Normalmente representan comportamientos específicos generados para una industria o compañía en particular. Estos pueden ser muy valiosos, como los que se incluyen para la industria de los call centers, o en el caso de este trabajo, templates específicos para simulaciones de aeropuertos desarrollados por terceras compañías como IBM.

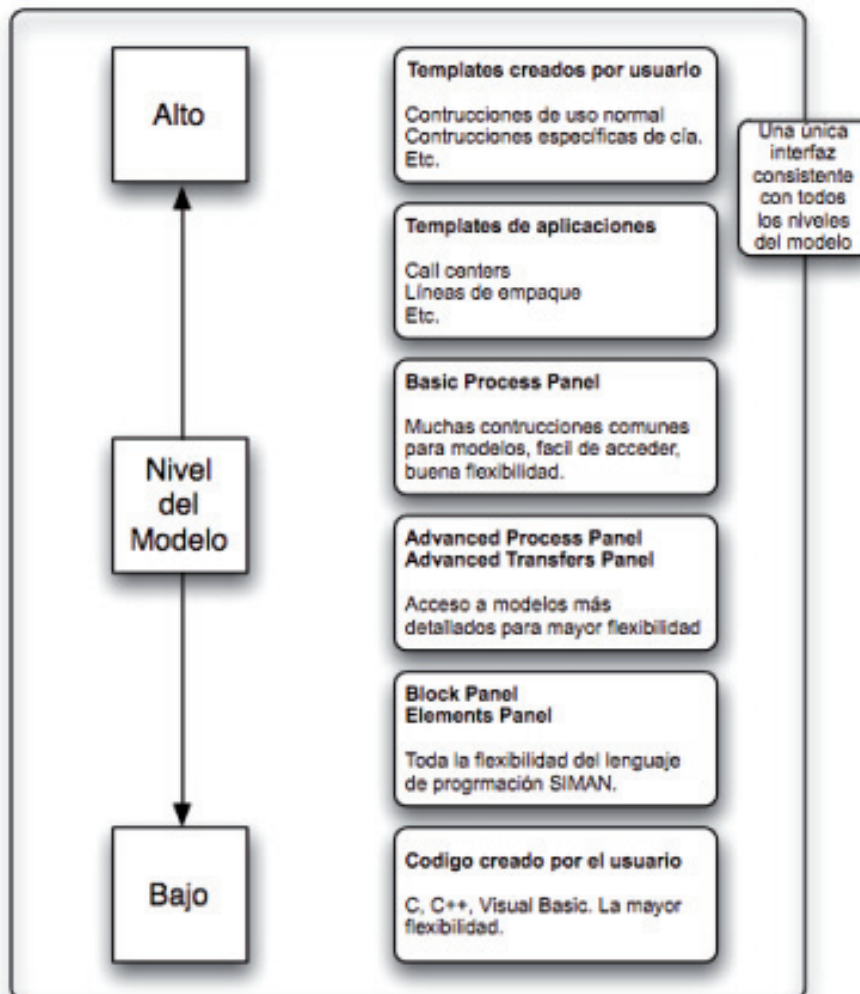


Figura 4 Estructura jerárquica de Arena. Kelton et al (2007)

Luego de los templates en la escala jerárquica de Arena, encontramos a los paneles. Estos contienen módulos con cierto nivel de complejidad que son comunes a gran cantidad de modelos y sirven como elementos básicos para la construcción de los mismos. El Basic process panel contiene módulos que permiten armar gran cantidad de modelos y está destinado a los usuarios que recién comienzan a utilizar Arena ya que están pensados para recolectar los comportamientos más utilizados en simulaciones en general sin necesitar demasiado desarrollo. El Advanced Transfers Panel y el Advanced Process Panel contienen en muchos casos, desagregaciones del Basic Process Panel, que permiten a los modeladores generar comportamientos más específicos para actividades “menos estándar”. En un nivel aún más bajo encontramos a los Blocks y los Elements, estos módulos son espejos directos del lenguaje SIMAN. Y por tanto son los que mayor flexibilidad de uso presentan, pero al mismo tiempo son los más complejos de utilizar y requieren de usuarios con mayores conocimientos.

Como mencionamos anteriormente, una de las grandes ventajas de Arena es la posibilidad de anexar código de lenguajes de programación tradicionales a nuestros modelos. Esta es la máxima libertad para un desarrollador, una combinación muy poderosa con un ambiente de alto nivel como el que estamos estudiando.

3.2. Modelos en Arena

Los modelos en Arena, como vimos en la sección anterior, pueden construirse en base a una gran cantidad de módulos de distinto valor funcional.

Un modelo está compuesto por distintos módulos, estos se clasifican, a su vez, en tres tipos, los procesos, los datos y la animación. Tomando instancias de cada uno de ellos como bloques o átomos constructivos podemos generar la lógica que necesitamos modelar. Debido a que Arena, como gran parte de los sistemas de simulación, utiliza lo que Thesen y Travis (1991) denominan flujo de transacciones (transaction flow) para representar los modelos, podemos generar los mismos en forma de diagramas de flujo con procesos unidos unos a otros describiendo el flujo de las entidades en su paso por el sistema. En las siguientes secciones analizaremos cada uno de los tipos de módulos que Arena utiliza.

Kelton et al (2007) recomiendan al usuario siempre tener una aproximación desde arriba hacia abajo a la hora de elegir los módulos a utilizar. Es decir, que se intenten utilizar primero los del Basic Process Panel y si no encontramos nada que se ajuste a nuestras necesidades, pasar al Advanced y al Blocks Panel.

3.2.1. Procesos

Los procesos (o Flowchart Modules, módulos de flujo) son los módulos de Arena que describen el comportamiento dinámico de los Modelos. A través de ellos, las entidades recorren el modelo realizando las operaciones detalladas. Según Kelton et al (2007) son los nodos del diagrama de flujo por los cuales las entidades pasan, se originan o salen del modelo.

Como figura en la sección 2.1.2. las entidades son los elementos del modelo que simulan los distintos tipos de unidades que se encuentran en el sistema y son afectadas por el mismo. Por ejemplo, si un sistema fuera un negocio de venta minorista y estuviese representado por un vendedor que puede atender a un cliente a la vez y el objetivo del sistema es atender clientes que entran, son atendidos y salen del local, los clientes serían las entidades (ver figura 5). Y para poder simular este sistema necesitamos generar clientes simulados que lleguen a este imitando la llegada de personas a un negocio. Es por eso que existen módulos que crean entidades. En el caso de Arena, el Create, parte del Basic Process Panel es el encargado de esta tarea. Este proceso va a generar entidades de tipo cliente para nuestro ejemplo tomando un criterio fijado en sus parámetros. Como lo que estamos haciendo es imitar al sistema real, necesitamos poder estimar el patrón de arribo de clientes del sistema real para poder generar los clientes simulados. O en caso de contar con información histórica de calidad, usarla para replicar eventos del pasado. El parámetro que controla el tiempo entre la creación de una entidad y la próxima, es fundamental en el módulo Create, ya que será el encargado de imitar el comportamiento del mundo real respecto al arribo de entidades al sistema. Generalmente, en los modelos que no usan directamente datos históricos para reproducir situaciones pasadas, se debe definir el patrón de arribo de entidades al sistema en forma de una distribución probabilística. El módulo Create usará esta para generar entidades. Para esto es clave poder analizar los datos históricos del sistema o tener acceso a realizar observaciones e investigaciones que nos permitan adaptar las muestras tomadas de la realidad a una distribución probabilística. Esta será usada para generar aleatoriamente los tiempos entre el arribo de una entidad y el arribo de la próxima. En la sección 3.5. veremos como Arena facilita el trabajo de encontrar una distribución apropiada para nuestro modelo.

Luego de creadas, las entidades atraviesan el modelo o diagrama de flujos ejecutando la simulación según los módulos o procesos que se encuentren en su camino. Estos módulos “atravesables” por las

entidades pueden tener distintos objetivos, pero básicamente están para dar al modelo los distintos comportamientos que imitan al sistema real. Un ejemplo es tomar un recurso, utilizarlo y liberarlo, como en el caso del vendedor, que es ocupado por un cliente mientras realiza la compra y lo libera al terminar; o módulos de decisión, que funcionan como un if-then-else de un lenguaje de programación; o módulos de transporte que imitan el movimiento de materiales en una industria, etc.

Una vez que las entidades recorren el modelo tienen que salir de él y ser destruidas. En este momento se aprovecha para registrar varias estadísticas sobre la entidad y ajustar los acumuladores estadísticos adecuados. Para esto existe el módulo Dispose que se encarga de recibir las entidades que deben salir de los límites del sistema.



Figura 5 Modelo simple de Arena

3.2.2. Datos

No todos los componentes necesarios para la lógica de un modelo pueden ser representados en el diagrama de flujos. Es por eso que existen otro tipo de módulos que permiten representar estos componentes. Para esto existen los módulos de datos, que se usan para representar algunos elementos de la simulación y algunas características de procesos. También sirven para definir variables y otros valores numéricos, como expresiones matemáticas y lógicas. Kelton et al (2007) dice que así como los procesos están plasmados en el modelo visualmente, los módulos de datos están “detrás de escena” haciendo un trabajo igualmente importante. En las próximas secciones describiremos en detalle los módulos de datos más importantes, pero para darnos una idea más concreta, estos pueden ser quienes definen las entidades y sus atributos, los recursos, las colas, y otros elementos. Otra diferencia con los procesos es que no puede haber más de una instancia de cada módulo de datos, y en lugar de eso, pueden albergar distintos datos en cada uno. Es decir, en el módulo colas vamos a encontrar todas las colas del sistema y ellas se relacionan con los módulos de procesos a través del nombre (tienen el mismo nombre).

3.2.3. Animación

La última, pero no menos importante, parte del modelo es la animación. A pesar de que ésta no cumple una función en la lógica o los resultados de la simulación, tiene razones fundamentales para existir. La animación da visibilidad al modelo, permite comunicar el proceso de simulación de manera mucho más efectiva que los diagramas y los datos. Es por eso que se usa para explicar y presentar el modelo a aquellos que no están vinculados a la producción del mismo, incluyendo a quienes encargan el modelo y se benefician con sus resultados. También quienes desarrollan el modelo usan la animación para entender el modelo y examinarlo rápidamente. En el proceso de verificación del modelo la animación es muy útil y permite trabajar juntos a los expertos en simulación y los expertos en el sistema real.

En el modelo, podemos agregar visibilidad a casi todos sus componentes a través de la animación. Las entidades pueden representarse por gráficos que pueden ser simples como círculos de colores o complicados como fotografías de lo que se representa. También pueden verse las colas que hacen las entidades a la espera de recursos o mostrarse el recurso con su representación gráfica y las modificaciones que realiza sobre las entidades. Además se pueden representar las variables para poder seguir la evolución de las mismas, así como generar gráficos a partir de ellas u otros datos o estadísticas que lleva el sistema. La animación es una herramienta muy potente y da versatilidad a la simulación y permite que ésta sea comprendida por una audiencia mayor.

3.3. Ejecución de una simulación

Una vez generado el modelo es muy simple correr la simulación en Arena, sin embargo por su flexibilidad existen varios aspectos a tener en cuenta y opciones a tomar dependiendo del objetivo que tenga la corrida.

Un aspecto clave a considerar a la hora de correr una simulación es la definición de una condición de terminación. Esto significa que si uno no especifica de manera explícita la condición o condiciones que deben darse para que el experimento termine, éste seguirá indeterminadamente. Estas condiciones tiene que ver con el objetivo del experimento y generalmente tienen que ver con el tiempo. Si se quiere saber

como afectará una nueva configuración de máquinas de una fábrica en un día tipo de trabajo, probablemente la simulación será de 16 horas o dos turnos de 8 horas. O puede ser que se busque entender el comportamiento durante una semana y ese será el tiempo simulado.

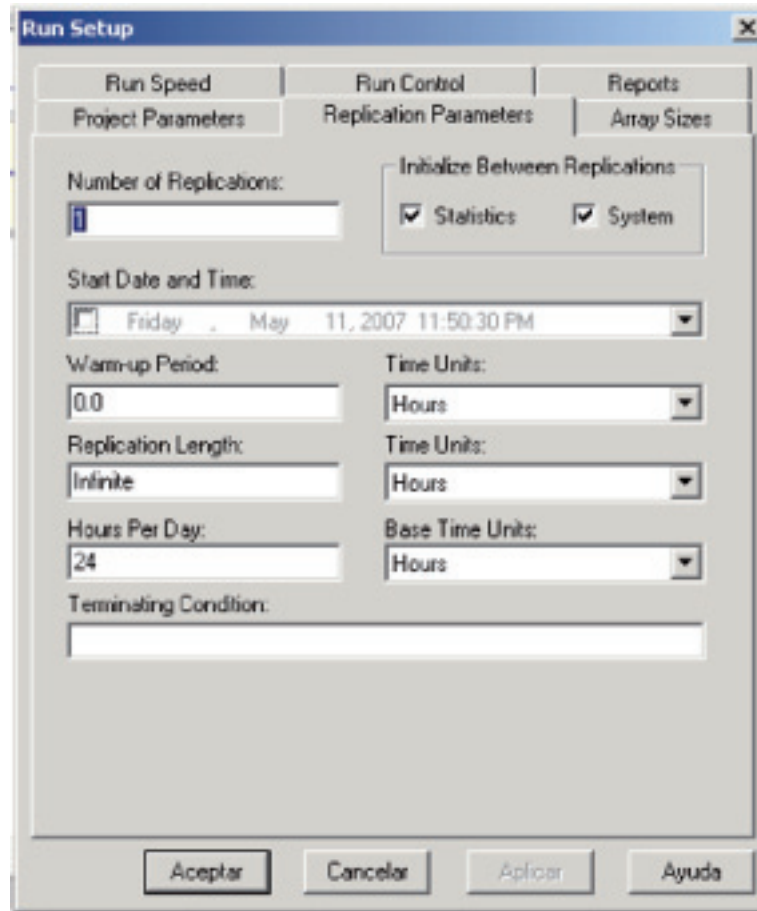


Figura 6 Parametrización de corridas en Arena

También existen situaciones en las que no es el tiempo el determinante del fin de una simulación, sino una condición del estado del sistema o sus variables, o el cumplimiento de una meta cuantitativa del proyecto. Por ejemplo, en una fábrica, hasta procesar N partes o en un call center hasta atender 100 llamadas. Siempre se debe, previo a correr una simulación, definirse la condición de terminación. Arena permite configurar el modelo para especificar estas condiciones de manera simple. En un menú creado a tal efecto (figura 6) se ingresan todos los parámetros respecto a la terminación de la misma, sea por tiempo o por una condición, que puede ser expresada como una expresión del lenguaje SIMAN o puede ser creada con la herramienta “expresion builder” de Arena, que permite usar una interfaz gráfica para generar expresiones lógicas y matemáticas.

Otro parámetro clave para correr un modelo es el número de réplicas que se harán del modelo. Anteriormente hemos discutido sobre cómo analizar los resultados de una simulación y el hecho de que como existe un factor aleatorio en la entrada de un modelo la salida tiene un componente aleatorio muy importante. Por esto, es muy importante que los resultados de una simulación sean analizados estadísticamente y no tomados como una verdad o como una observación del sistema real. Si estamos usando el modelo para tomar una decisión sobre el resultado de correr el experimento una vez, probablemente cometamos un error, ya que éste puede estar afectado por valores que salen de lo usual del sistema real. En cambio, lo recomendable es realizar un número de simulaciones y analizar los resultados de todas las simulaciones con técnicas estadísticas para obtener medias y promedios que se aproximen a la realidad del sistema y nos ayuden a evitar los sesgos que una u otra corrida puedan mostrar debido a los efectos de la generación de números aleatorios en las entradas. Arena prevé que se ingrese la cantidad de réplicas deseadas del experimento. Permite también, recolectar estadísticas en cada réplica de forma independiente o mantener el estado del sistema y las variables al fin de cada una para acumularlas y pasarlas al comienzo de la próxima obteniendo una serie de experimentos concatenados entre sí. Esto es importante y se ve en la figura 6 como los dos tildes de “System” y “Statistics” encuadrados

en inicializar réplicas. Existen razones para dejar uno o ambos tildes vacíos, pero para que los resultados de la simulación de cada réplica sean estadísticamente independientes y distribuidos idénticamente (IID) debemos tildar ambas opciones. Recordemos que ésta es una condición necesaria para poder analizar estadísticamente los resultados usando técnicas estándar de inferencia estadística. En esta etapa, Arena también permite que el usuario decida, en otra pestaña del menú de la figura 6, elegir de qué elementos del modelo se tomarán estadísticas, según las necesidades, puede recoger datos de entidades, recursos, colas, etc. Desde este menú también se eligen las unidades de tiempo en las que se va a medir todo el sistema para hacer referencias (ver figura 7).

Luego de elegir las condiciones de la simulación hay que correr el experimento propiamente dicho. Para esto existen una serie de opciones según el objetivo de la corrida específica. Luego de armar un modelo, al igual que un programa escrito en un lenguaje tradicional, hay que chequear si el modelo corre sin errores. Para eso podemos usar la opción "Check" del menú "Run" (ver figura 3) que recorre el modelo y nos avisa si hay alguna inconsistencia o error. Luego de esto estamos verdaderamente listos para correrlo. Y podemos usar una de las siguientes opciones del menú "Run" (ver figura 3):

Usar el botón "Run" para correr la simulación. En este caso se hace un chequeo automático antes de correr y mientras se realiza la simulación, la animación muestra su progreso.

"Run/Step" ejecuta la simulación un evento a la vez para poder seguir su evolución y, generalmente, buscar y encontrar errores en el modelo.

"Run/Fastforward" corre la simulación con la animación deshabilitada a una velocidad mayor y permite detenerla para ver la animación en cualquier momento. También chequea el modelo antes de correrlo.

"Run/Pause" pausa la ejecución del modelo para ver algún aspecto específico, puede continuarse luego.

"Run/StartOver" sin importar en qué momento de la simulación nos encontremos, hace que todo vuelva a empezar.

"Run/Review errors" muestra los errores detectados en el último chequeo para permitir solucionarlos.

"Run/Run Control/Command" abre una ventana de línea de comandos y corre la simulación mostrando sus valores y permite modificarlos durante la ejecución.

"Run/Run Control/Break" Igual que el anterior, pero permite elegir un momento predeterminado para frenar la ejecución.

"Run/Run Control/Highlight Active Module" hace que el módulo que se está ejecutando del diagrama sea iluminado.

"Run/Run Control/Batch Run" es la forma más rápida de correr el modelo sin animación.

"Run/SIMAN" permite ver y modificar los archivos de código SIMAN generados por Arena.

De esta manera Arena permite la mayor versatilidad a la hora de correr un modelo, sea para ver resultados o mejorarlo.

3.4. Reportes de una simulación

En Arena, al final de una simulación se genera un reporte con los resultados de la misma, éste es accesible desde la barra de proyecto en la pestaña "Reports" (ver figura 3). Haciendo un clic allí se descubre el árbol jerárquico que contiene la tabla de contenidos del reporte. Durante la simulación, Arena genera una base de datos con todos los datos de la misma, lo hace en formato Microsoft Access y un archivo con el nombre del modelo y la extensión .mdb es la salida de este proceso. El usuario puede guardar este archivo y volverlo a usar con Microsoft Access para recalcular valores o guardar la historia del experimento. El reporte de la simulación de Arena se genera tomando ese archivo con una herramienta muy utilizada en el mercado llamada "Cristal Reports", que permite gran versatilidad y una excelente presentación de los mismos. Además, esto permite que los reportes sean exportados a los formatos más usados de procesadores de texto y planillas de cálculo.

Hay varios parámetros a fijar para que Arena, al correr la simulación, tome todos los datos necesarios de aquellos módulos y partes del modelo que nos interesan para obtener resultados.

En la figura 7 vemos el menú de configuración de la corrida de la simulación donde debemos marcar aquellos módulos de los que queremos que se recolecten los datos. Como vemos, tenemos muchas partes estándar de todo modelo en este menú, tales como: entidades, recursos, colas y procesos.

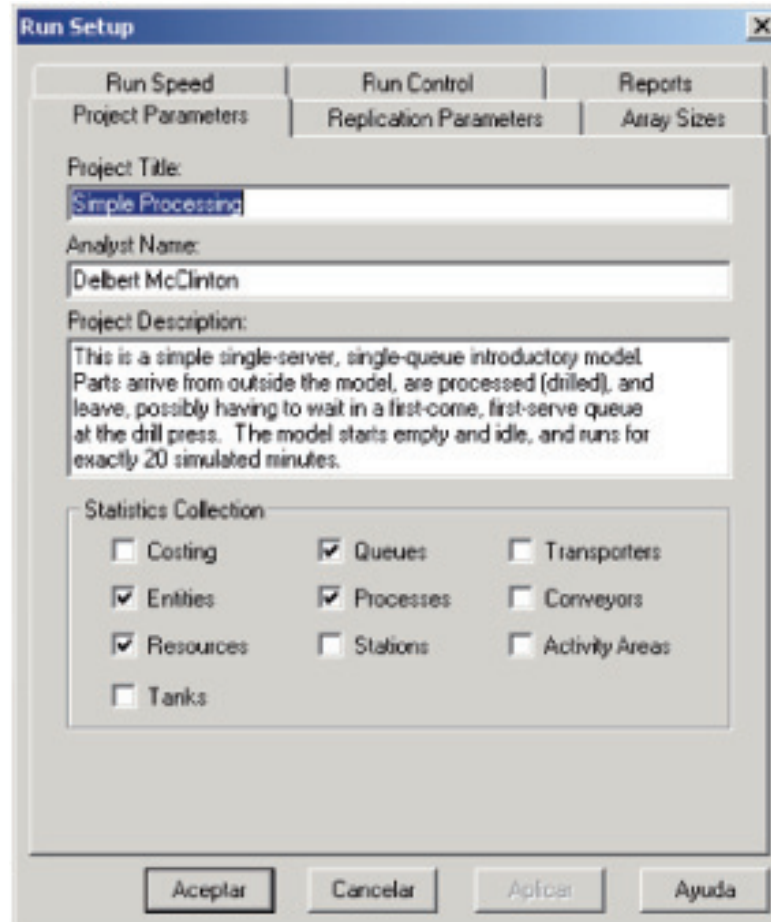


Figura 7 Elección de recolección de estadísticas en Arena

También existen otras más específicas de modelos de industrias o tipos particulares como: costeo, tanques, estaciones, transportes, cintas transportadoras y áreas de actividad. Las únicas estadísticas generadas que no están este menú son las que son estándar en Arena y que el usuario genera agregando módulos de tipo "Record" en el modelo.

Para cada uno de estos elementos del modelo, para los que Arena genera estadísticas, se producen una serie de cálculos y resultados que se pueden clasificar en "Tally Statistics" o estadísticas de tiempo discreto, "Time-Persistent Statistics" o estadísticas de tiempo continuo y "Counters Statistics" o contadores.

Las Tally Statistics son aquellas que resultan de tomar promedios, máximos y mínimos de listas de números. Un ejemplo de este tipo de resultados son los vinculados a las entidades, como tiempo promedio pasado en el sistema, que surge del promedio de los tiempos de todas las entidades que pasaron por el sistema durante la simulación.

Las Time-Persistent Statistics son aquellas que surgen de tomar tiempos promedio, mínimos y máximos de variables y elementos en los que debe computarse el tiempo. Es el valor determinante de los resultados y éste debe ser computado como una variable continua. Ejemplos de este tipo de estadísticas son: promedio de entidades esperando en una cola, utilización de recursos, etc.

Por último los contadores o Counter Statistics, son aquellos que como su nombre lo indica son acumuladores de variables relativas al sistema. Algunos ejemplos son: tiempo total acumulado, cantidad de entidades procesadas por el sistema, etc.

A pesar de que recién hemos visto la clasificación de los resultados que genera una simulación, la organización de todas estas estadísticas en el reporte que genera Arena es otra. En primer lugar se presenta un reporte llamado "Category Overview" (ver Anexo C) donde están todos los datos de salida de la simulación resumidos y calculados para todas las réplicas de la simulación llevadas a cabo. En su interior encontramos datos de todos los elementos de la simulación, que en su caso más complejo serían:

Key performance indicators: aquí encontramos resumidos datos relevantes del sistema como entidades, recursos y sistema en general.

Activity area (level n): Arena permite definir Activity Areas en un sistema, esto permite agrupar a los elementos del sistema en subsistemas con comportamientos u objetivos en común. Es por eso, que si

el sistema está definido de esta manera se generan reportes resumiendo tiempos y costos de todo lo relacionado con estos subsistemas.

Conveyor: En caso de que nuestro modelo sea de un proceso industrial que incluya cintas transportadoras, el reporte genera estadísticas particulares para estos módulos. A diferencia de otros elementos del modelo, éste solo presenta estadísticas sobre su uso.

Entity: Aquí encontramos todas las estadísticas de las entidades del sistema resumidas y calculadas sobre todas las réplicas simuladas. Estas son tiempos, costos y contadores de entrada y salida.

Process: Los resultados calculados para los procesos del modelo son tiempos por tipo de entidad, tiempo acumulado, costo por entidad, costo acumulado y contadores de entidades entrantes y salientes.

Queue: O colas, incluye, tiempo de espera, costo de espera y número de entidades en espera.

Resource: Para los recursos se obtienen datos sobre uso y costos.

Transporter: Ésta estadística solo se calcula para aquellos modelos que incluyen sistemas de transportes. Para ellos se calculan datos sobre su uso.

Station: Las estaciones en Arena están relacionadas al concepto de Activity Areas, ya que se usan para definir los límites de estas áreas. Es por eso que para ellas solo se muestran datos sobre el número de entidades que transitaron por allí.

Tank: Es un módulo por el cual Arena agrega la posibilidad de modelar actividades continuas que salen de los modelos discretos estándar. Para estos módulos se calculan estadísticas de uso.

User Specified: Aquí se muestran las estadísticas que no son estándar a Arena y fueron puestas en el modelo por el usuario mediante los módulos "Statistic" y "Record" y se muestran divididas en "Tally", "Counters", "Time-Persistent" y "Output". Los primeros fueron mencionados antes, y "Output" muestra aquellas expresiones generadas explícitamente por el usuario en los campos con el mismo nombre de los módulos y es la forma más flexible de generar datos a medida en la salida de una simulación.

Luego de este encontramos los siguientes reportes: "Category by Replication", que contiene los mismos datos que el anterior, pero en lugar de tener el resumen de todas la réplicas, muestra los datos de cada una de ellas por separado; luego de este encontramos un reporte pormenorizado de todos los datos de cada elemento del modelo "Activity Areas", entidades, procesos, colas, recursos, "Transfers" y "User Specified". Internamente, estos reportes están ordenados por réplica.

De cada uno de los datos generados por Arena para su reporte se calculan el promedio, el máximo, el mínimo y el desvío para un intervalo de confianza del 95%. Para que este último dato pueda ser calculado por Arena se debe correr más de una réplica del modelo, de otra forma, si sólo corremos una réplica, el reporte estará completo con todos los datos salvo ese, donde tendremos un mensaje que dice insuficiente.

El dato más importante en cada caso, que nos servirá para conocer el comportamiento del sistema modelado y que, en la mayoría de las oportunidades, nos servirá de base para contestar las preguntas que nos llevaron a realizar el experimento de simulación, es sin duda el promedio por sobre máximos y mínimos. Por otra parte, el intervalo de confianza nos servirá para tener una estimación de la precisión del promedio respecto al nivel de aleatoriedad del modelo comparado con el sistema real.

Como mencionamos antes, Arena tiene un comportamiento particular respecto al intervalo de confianza del 95% que presenta. Ya que sólo lo hace si hacemos más de una réplica de la simulación. Esto tiene que ver con qué es y cómo se calcula el intervalo de confianza en Arena. Una observación bastante obvia es que un valor promedio de una simulación se calcula para conocer cómo el sistema real modelado se comporta respecto a ese valor y la idea de que si realizamos más de una réplica de la simulación, dado que tenemos valores aleatorios en la entrada ese valor cambia entre una y otra. Entonces, si tomamos el valor promedio entre las réplicas tendremos un valor más cercano al "valor típico" generado por el sistema real, y que mientras más réplicas tengamos, el promedio se hará tomando más valores y por ende será aún más cercano al generado por el sistema real. Esto se relaciona con el intervalo de confianza, ya que mientras más réplicas estemos tomando para nuestro experimento éste será más pequeño. Esta noción proviene de la idea de que el promedio esperado de un valor (normalmente denominado μ) es aquel que surge del promedio de los valores generados por un número infinito de réplicas, y que por esto, este valor no tendrá ningún grado de incertidumbre asociado. Pero como realizar infinitas réplicas es inviable, el intervalo de confianza de 95% es aquel que nos permite pensar que existe un 95% de probabilidad de que ese valor esperado μ estará contenido en el intervalo. El 5% restante es el grado de incertidumbre y mientras el intervalo sea más pequeño nos dará más seguridad sobre el uso del valor promedio. Según Kelton et al (2007) "...un intervalo de confianza nos da un estimado de μ , así como una idea de cuán preciso es este estimado."

Todo esto está basado en cálculos que realiza Arena utilizando teoría de estadística clásica y por eso es fundamental que las réplicas realizadas en el experimento sean IID.

3.5. Análisis de entradas con Arena

Para desarrollar un modelo en Arena debemos realizar el diagrama de flujos que representa al modelo en sus aspectos estructurales y funcionales, es por eso que a ésta parte del desarrollo se la llama modelado estructural (structural modeling), pero también existen otros elementos en el modelo que deben ser definidos para obtener una representación fiel del sistema real. Estos otros elementos suelen ser de naturaleza matemática o numérica y por eso a esta parte del desarrollo se la denomina modelado cuantitativo (quantitative modeling), y es tanto o más importante que el modelado estructural.

Entre las partes más importantes de esta última actividad encontramos la definición de cómo es el proceso de arribo de entidades al sistema o de cómo se comportan los recursos respecto al tiempo que tardan en realizar una tarea, o cuál es el nivel de servicio de un recurso y por ende cómo es su patrón de fallas. Todos estos aspectos son clave en un modelo, ya que son estos quienes van a determinar si el comportamiento del mismo se parece o no, y cuánto, al sistema real.

Para poder realizar esta tarea, el primer paso es siempre buscar datos del sistema real. Ya sean datos históricos o documentos de especificaciones si es que el sistema no ha sido construido.

En el caso de que hayamos tenido la suerte de conseguir datos históricos debemos decidir si vamos a nutrir nuestro modelo con ellos directamente (modelo determinístico) o vamos a buscar una distribución probabilística que se ajuste lo suficiente a estos como para ser usada en su lugar para generar datos aleatorios (modelo estocástico). Según Kelton et al (2007) el hecho de usar números aleatorios en un sistema es siempre más realista y además puede llevar a descubrir comportamientos importantes en el modelo. Es decir, el hecho de que existan datos históricos y que estos hayan ocurrido en el sistema real, no significa que estos no estén sesgados de alguna manera y que sirvan para estudiar el modelo de manera completa. Es por eso que normalmente los modelos se generan usando datos aleatorios, pero no por esto hay que descuidarse, ya que si estos datos no son lo más representativos de la realidad posible, los resultados de la simulación también van a estar lejos de la realidad y cualquier decisión tomada con ellos será, muy posiblemente, un error.

Para ayudarnos en esta tarea, Arena provee el "Input Analyzer" que mediante el uso de datos históricos nos provee de una distribución probabilística adecuada para su uso en nuestro modelo. Esto se puede hacer mediante la comparación de los datos con posibles distribuciones para encontrar la más adecuada, o elegir explícitamente una distribución y hacer que el programa nos de los parámetros (media, mediana, máximo, mínimo, etc.) de la misma que más se ajustan a los datos provistos. En cualquier caso, el "Input Analyzer" nos da una expresión que podemos usar directamente para configurar los procesos del modelo mediante una operación de copiado y pegado.

Las distribuciones probabilísticas pueden ser clasificadas en dos categorías fundamentales, las teóricas y las empíricas. Las primeras están basadas en funciones matemáticas, mientras que las segundas dividen los datos de la muestra en grupos y calculan los valores de proporción en cada grupo. Asimismo, estas pueden ser divididas en discretas y continuas. Las continuas tienen un valor de salida para cada valor del conjunto de los números reales de la entrada. Las discretas solo tienen valores de salida para entradas del conjunto de los números enteros. Arena permite cualquier combinación de estos tipos en la distribución a generar.

Para poder usar el "Input Analyzer" primero se debe crear un archivo de texto con los datos "crudos" de las observaciones históricas que tenemos. Este archivo es usado por el programa para generar un histograma con las muestras para que procedamos a encontrar nuestra distribución. Generalmente es recomendable usar la opción "Fit All" del programa que directamente compara nuestros datos con todas las distribuciones teóricas dando medidas cuantitativas de cómo se ajustan nuestros datos a cada distribución.

Es muy difícil saber cuando estos valores son aceptables, ya que muchas veces son muy parecidos entre sí y ninguno es determinadamente claro por lo cercano a los límites mayores. Entonces debemos optar por seguir buscando cuál de las distribuciones teóricas está más cerca de nuestras necesidades o pasar a generar una empírica. Para tomar esta decisión no existe una regla dada, pero durante el proceso de comparación y ajuste de los datos obtenidos con las diversas distribuciones podemos advertir si alguna de las teóricas es lo suficientemente ajustada como para usarse en lugar de nuestros datos. Si esto no ocurre será mejor optar por una empírica, para lo cual podemos seleccionar una discreta o una continua. En caso de haber optado por una teórica y que hayan varias que tengan valores similares de las pruebas de ajuste a nuestros datos, hay algunos aspectos a considerar para poder realizar nuestra elección. En primer lugar, debemos seleccionar entre las distribuciones posibles teniendo en cuenta si éstas tienen límites o no, es decir, si una de ellas puede arrojar valores para cualquier entrada hasta el infinito, aunque sean los valores con menor posibilidad de aparecer, si lo hacen y esto es un comportamiento imposible de observar en el sistema real, debemos inclinarnos por la que tiene límites. En segundo lugar, debemos considerar los parámetros de las distribuciones, hay algunas que son fáciles de comprender y por ende

configurar y otras que son más complejas, es preferible elegir la menos compleja. Por último, podemos hacer un simple análisis de sensibilidad corriendo el modelo con ambas distribuciones para ver si los resultados son significativamente distintos.

3.6. Basic process panel

Como vimos en el comienzo de la sección, el Basic Process Panel, o Panel de Procesos Básicos, contiene los módulos de Arena con el mayor nivel de abstracción, y se recomienda que sean nuestra primera elección a la hora de armar nuestro diagrama de flujo por sobre los módulos de otros paneles. En el Anexo A de este trabajo encontraremos una descripción pormenorizada de cada módulo de este panel.

3.7. Advanced process panel

El Advanced Process Panel, o Panel de Procesos Avanzados, contiene módulos de un nivel de abstracción menor en su funcionalidad que los del Basic Process Panel. Sin embargo, muchas veces estos módulos brindan gran flexibilidad a la modelización de sistemas y según Kelton et al (2007), son la elección más usual de muchos usuarios de Arena experimentados. En el Anexo A de este trabajo encontraremos una descripción pormenorizada de cada módulo de este panel.

3.8. Otros paneles avanzados

Además de los paneles vistos en las secciones anteriores existen otros destinados a implementar acciones de industrias o comportamientos específicos en Arena. A pesar de ser menos usados que los anteriores y destinados a problemas particulares y usuarios de Arena más experimentados, los módulos que vamos a estudiar en esta sección son muy importantes. Podemos dividirlos en dos niveles de complejidad y abstracción. Por un lado tenemos al "Elements Panel" y al "Blocks Panel" que representan el nivel más bajo de abstracción en Arena. El primero tiene módulos que representan las construcciones en SIMAN de todos los elementos que pueden hallarse en un modelo, y que sirve para acceder al nivel más primitivo de ellas en caso de que sea necesario modelar un comportamiento fuera del estándar para el cual fueron programados. El segundo tiene módulos que representan las sentencias de SIMAN y permite máxima flexibilidad en los modelos sin la necesidad de codificar en lenguaje SIMAN. Por otro lado tenemos a los paneles que incluyen modelos de un nivel de abstracción un poco más alto y que dan nueva funcionalidad y flexibilidad respecto al Advanced Process y al Basic Process. El Advanced Transfers Panel tiene módulos destinados a modelar todo tipo de modelos que incluyan transportes de algún tipo, para eso, incluye tres grupos de módulos, el general que incluye la casuística de la mayor parte de movimientos industriales y logísticos; el de cintas transportadoras, que permite agregar el comportamiento detallado de éstas; y el de transportes, que da aún más flexibilidad a los movimientos de entidades en rutas y áreas definidas. El Flow Process Panel trae módulos necesarios para representar modelos continuos en Arena. Permiten representar flujos continuos de movimiento de líquidos u otros materiales que tienen comportamientos similares.

Además de estos paneles, Arena, tiene una edición especial hecha para la industria de los Call Centers, sobre la cual se pueden representar con facilidad y versatilidad este tipo de servicio y toda su problemática. Otra edición especializada de Arena está enfocada en las líneas de empaque de industria, para la cual hay elementos particulares y comportamientos específicos que son capturados por esta edición.

3.9. Comparación de escenarios en Arena

En muchos casos las simulaciones son realizadas para ayudar a diseñar un sistema nuevo o solucionar un problema de uno existente. Es en estos casos, en los que normalmente se desarrollan varias alternativas del sistema, ya sea en su lógica o en su configuración, se realizan experimentos de simulación sobre los distintos escenarios para predecir el comportamiento de cada uno y elegir el mejor para ser implementado.

Para poder realizar esto tenemos varias opciones, la primera, que podemos realizar con los procedimientos descritos hasta ahora en el presente trabajo, sería generar el modelo, correrlo, exportar los resultados a alguna planilla de cálculo y luego modificar el modelo para repetir la operación y llevar a cabo la comparación de los resultados en algún programa externo con la complejidad que implican los estudios estadísticos de los mismos. Arena provee maneras de automatizar este trabajo a través de herramientas que se integran con la aplicación y permiten usar la información y modelos generados para realizar las comparaciones.

El primer método provisto por Arena para esto se encuentra en una aplicación adjunta llamada "Output Analyzer", en ella encontramos una forma fácil de comparar hasta dos escenarios. Esta requiere de algunos ajustes en el modelo para poder recolectar la información que este programa usará para hacer las pruebas necesarias y ayudarnos a llegar a una conclusión respecto a ellos. Debemos utilizar módulos

“Statistic” en el modelo para recolectar información de los parámetros que deseamos comparar generando archivos binarios .dat sobre los mismos. Luego podemos usar esos archivos como entrada del “Output Analyzer”, donde indicaremos cuáles son los parámetros de cada modelo, que incluirá información sobre todas las réplicas realizadas de cada escenario distinto. Una vez que ingresamos los datos, el programa realiza pruebas estadísticas para determinar si los resultados provistos difieren entre sí realmente o si la diferencia que se observa a simple vista es el resultado de “ruido estadístico” relacionado con la generación y el uso de números aleatorios. Dándonos como resultados intervalos de confianza de 95% sobre las diferencias esperadas en los valores, y si estos intervalos no cubren el valor cero, podemos concluir que existe diferencia estadística real entre los valores. Además, para ratificar el resultado presenta una prueba de hipótesis sobre los valores, para la cual un “no rechazado” confirma lo supuesto a partir de que el intervalo de confianza esta fuera de rango del valor cero. De esta manera podemos concluir a través de una prueba estadísticamente válida, a contrario de observaciones empíricas, si los resultados son confiables o no.

Otra forma provista por Arena para realizar comparaciones de escenarios de modelos es a través de otro programa llamado “Process Analyzer” (o PAN para abreviar). Este programa, a diferencia del “Output Analyzer” utiliza como entrada modelos de Arena en lugar de archivos con datos de salida de simulaciones. Cuando corremos o chequeamos un modelo, Arena genera un archivo .p, algo así como si compilara el modelo e hiciera un ejecutable, comparando con la programación general. El PAN toma como entrada dichos archivos, pero para que podamos utilizarlos debemos recordar que si queremos probar distintas configuraciones de un modelo, debemos dejar los parámetros a cambiar en cada escenario en forma de variables de Arena o cantidades de recursos, ya que estos son los únicos que el PAN toma para hacer los experimentos. En la figura 8 vemos una pantalla luego de realizar una comparación.

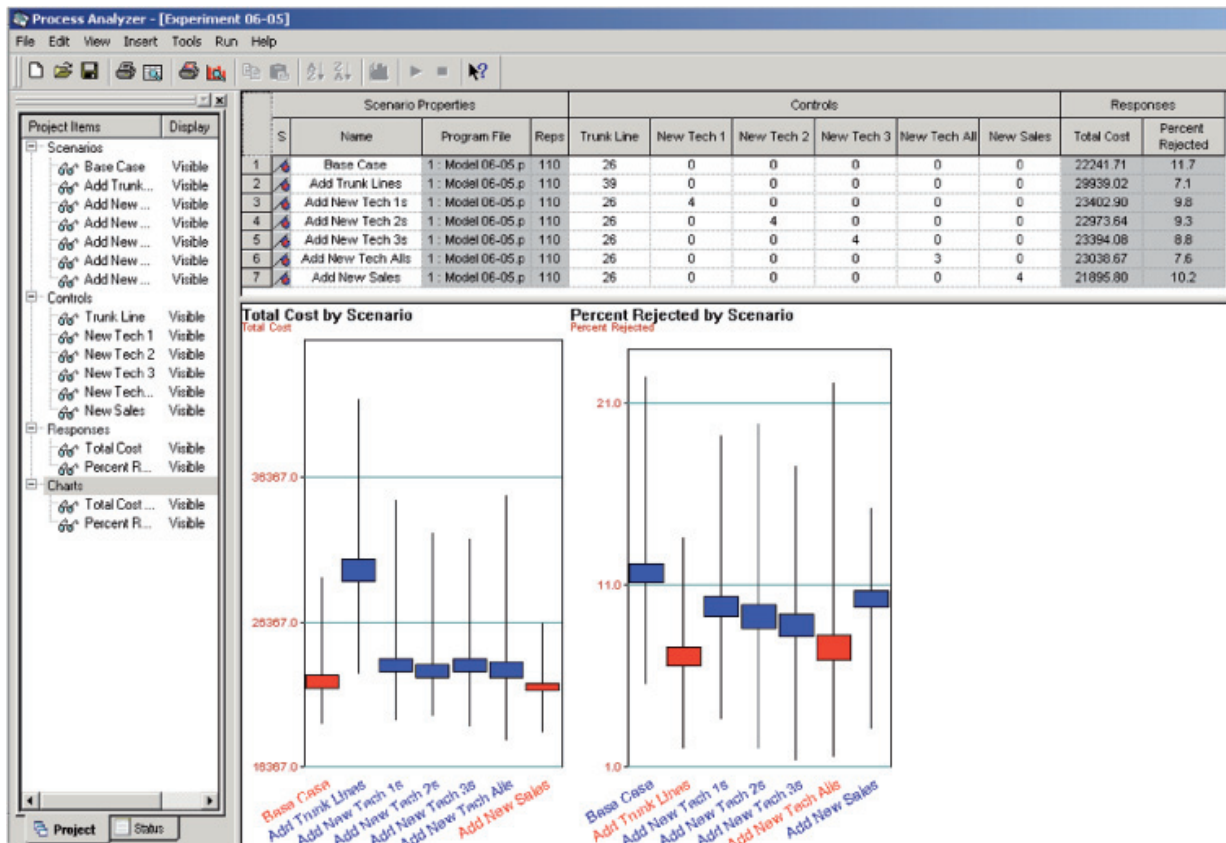


Figura 8 Pantalla del Process Analyzer (PAN)

A la derecha arriba vemos como seleccionamos los escenarios. Nombre, el archivo .p a utilizar y la cantidad de réplicas son los datos a ingresar en esta sección, luego, en la sección de controles están las variables y las capacidades seleccionadas para modificar la configuración del modelo en cada escenario. Una vez completados los datos, podemos correr la simulación para obtener resultados. Estos son presentados en la tabla a la derecha de la pantalla y en el gráfico en la sección inferior de la misma. La tabla muestra los promedios de todas las réplicas de los parámetros seleccionados como resultados de la simulación. Y el grafico de “bigotes” muestra los valores mínimos y máximos, junto con el promedio

y su intervalo de confianza de 95% en forma de rectángulo para cada escenario. Los resultados en rojo son los mejores de cada experimento.

Como vemos Arena permite realizar comparaciones válidas con gran flexibilidad y versatilidad dando opciones para cubrir distintos tipos de casos posibles.

3.10. Búsqueda de un escenario óptimo

En la sección anterior vimos cómo a partir de escenarios creados por el usuario podemos compararlos y obtener conclusiones sobre la diferencia de los mismos. O sea, podemos darnos cuenta de cuál de los escenarios es mejor para nuestro objetivo, pero no sabemos si existen mejores en el universo de posibles escenarios.

Para esto, Arena provee una herramienta que usa técnicas avanzadas de búsqueda en el espacio de posibles configuraciones del sistema para encontrar una cuasi óptima. Normalmente, cuando se desarrolla un modelo para simular distintos escenarios, encontramos una serie de parámetros a configurar en mismo. Estos son los que tomarán distintos valores para dar lugar a los nuevos escenarios. En general, estos parámetros están compuestos por una serie de variables del modelo. El hecho de que sean varios los valores a modificar implica que si quisiéramos probar todas las combinaciones posibles para probar todos los escenarios posibles de ese modelo, y elegir el que mejores resultados presenta, probablemente nos encontremos con tantas alternativas (hasta millones) que será imposible de realizar. Es por eso que el trabajo de estas técnicas de búsqueda heurística, que en el caso de Arena son Búsqueda Tabú o Búsqueda Dispersa (Scatter Search), permiten encontrar una configuración que de una respuesta cuasi óptima buscando eficientemente sin la necesidad de tomar el tiempo computacional necesario para evaluar todas las posibles opciones.

Dentro del grupo de programas que componen Arena está uno llamado OptQuest (por las palabras en inglés de búsqueda de óptimo) que realiza este proceso de búsqueda heurística. Así como el PAN, éste trabaja con el modelo mismo de Arena. También debemos definir cuáles son los "controles" (variables o recursos) que determinan una configuración del mismo y cuáles son los parámetros de salida por los cuales será evaluado. Una vez hecho esto, debemos poner algunas restricciones sobre los controles, para acotar el espacio de búsqueda, y especificar de qué manera se vuelve óptimo el resultado. Por ejemplo, minimizando o maximizando un resultado. Con todas estas condiciones el programa comienza a realizar la búsqueda generando automáticamente configuraciones y realizando las simulaciones para obtener los resultados, avanzando hacia los mejores en busca del óptimo. El sistema detiene automáticamente la búsqueda cuando considera que los sucesivos cambios que se generan en la configuración no mejoran sensiblemente el resultado de la simulación. Así la configuración que se obtiene no va a tener garantías de ser la óptima, pero sin duda estará entre las mejores que puede dar el sistema.

4. Simulación de colas en un aeropuerto

En esta sección presentaremos el marco teórico específico en el que se inserta el problema a resolver, utilizando Arena para realizar el modelo y la simulación.

4.1. La utilización de la Simulación en el análisis de la capacidad de los aeropuertos

Los aeropuertos son generalmente grandes obras de infraestructura que demandan importantes sumas de dinero para su realización. De acuerdo con Doshi y Moriyama (2002)⁵ la renovación del aeropuerto internacional de Toronto en Ontario, Canadá, demandó casi 4.500 millones de dólares americanos. Por lo dinámico de la industria respecto a cambios de tecnología y legislación, así como de cambios permanentes en la demanda de servicios; es muy importante para quienes construyen y financian aeropuertos, que dichos proyectos sean efectivos a la hora de su operación. Por estas razones encontramos muchos estudios realizados en predecir los comportamientos de los agentes que interactúan en estas obras para poder entenderlos y dimensionar los proyectos de manera acorde.

Como se explica en la sección 1, podemos dividir el aeropuerto en subsistemas que se relacionan entre sí dando como resultado el gran sistema aeroportuario. Estas partes del proceso general presentan distintas necesidades para su funcionamiento y requieren de una especial atención al detalle para ser estudiadas. Entre ellas encontramos la pista con toda la complejidad que su manejo acarrea, las posiciones de embarque que deben tener un planeamiento minucioso, los flujos de pasajeros que son claves ya que el pasajero es considerado el cliente más importante en un aeropuerto, los equipajes cuyo manejo en tiempo es muy importante, y los sistemas de transporte de vehículos terrestres entre otras. Como podemos observar existen una gran cantidad de subsistemas que pueden ser estudiados mediante distintas técnicas.

5. Doshi, N., Moriyama, R.: "Application of simulation models in airport facility design", Proceedings of the 2002 Winter Simulation Conference

Para estudiar todos éstos podemos usar técnicas que pueden ser aplicadas a una u otra parte del sistema. Janic (2000)⁶ enumera una serie de técnicas basadas en modelos analíticos para el análisis de ciertos subsistemas. Entre ellas encontramos teoría de colas, teoría de flujo de redes (network flow theory) y teoría de redes de colas (theory of queueing networks). Sin embargo, tanto Janic (2000) como Joustra y Van Dijk (2001)⁷ encuentran que la simulación es la mejor forma de estudiar aeropuertos por sobre las nombradas anteriormente. Por un lado, la teoría de colas, que es la más utilizada luego de la simulación, es muy rígida y no permite modelar los comportamientos de picos en la demanda que se generan en el sistema real. Por otro lado, la simulación permite una versatilidad de modelado que no sólo incluye los problemas o sistemas que podrían ser modelados con teoría de colas, sino que permite incluir la gran mayoría de los comportamientos encontrados en un aeropuerto sin tener que generar modelos heterogéneos, es decir bajo el mismo ambiente de simulación. Como último componente decisivo, el modelo generado puede ser guardado y modificado las veces que se requiera en función de cambios observados en el sistema real, que en esta disciplina encontramos de manera tan frecuente. En su estudio del aeropuerto de Schiphol, en Ámsterdam, Holanda, Gatersleben y van der Weij (1999)⁸ enumeran las siguientes razones por las cuales eligieron la simulación como herramienta para analizar los flujos de pasajeros en los diseños de una modificación al aeropuerto: a) interdependencia de procesos, el hecho de que los procesos en un aeropuerto estén dispuestos en serie genera una dependencia mutua que puede ser modelada, pudiendo entender como la modificación de una impacta en el sistema general; b) el comportamiento dinámico de los agentes en un aeropuerto así como las condiciones cambiantes en el mismo pueden estudiarse mediante la simulación, permitiendo encontrar la combinación justa de organización de procesos e infraestructura, inclusive para horas pico; c) el hecho de poder generar distintos escenarios nos permite no solo evaluar posibles soluciones para encontrar la mejor, sino ser proactivos analizando hipótesis de cambios futuros y su impacto en el sistema.

Encontramos muchos ejemplos de estudios en los cuales se aplica simulación a problemas relacionados con aeropuertos. En su trabajo, Verbraeck y Valentin (2002)⁹ dicen: "*Los aeropuertos son un área ideal para la aplicación de simulación*". Joustra y Van Dijk (2001) consideran necesario el uso de simulación para el estudio de los procesos relacionados al check-in en los aeropuertos. También Doshi y Moriyama (2002) y Kiran et al (2000)¹⁰ usan una aplicación de simulación a la evaluación del diseño de un aeropuerto completamente nuevo. Takakuwa y Oyama (2003)¹¹ realizan un estudio de simulación para intentar dar mayor eficiencia al aeropuerto de Kansai en Japón. Por su lado, Pendergraft et al (2004)¹², explican cómo se utilizó la simulación para determinar el impacto de las medidas de seguridad introducidas en los aeropuertos de EE.UU., a partir de los incidentes del 11 de septiembre de 2001, y cómo también se utiliza de manera estándar en dicho país para dimensionar recursos de aeropuertos y generar políticas y reglamentos sobre diseño, seguridad y niveles de servicio a dar en las terminales aéreas. También, de manera de ratificar la utilidad y la asiduidad de uso de simulación en casos de aeropuertos, encontramos sistemas OTS (off the shelf) que están hechos para simular cualquier tipo de aeropuerto provistos por organizaciones como IATA, IBM y otras de renombre.

Hemos podido comprobar que la simulación es una herramienta efectiva y de uso común en el análisis de aeropuertos en sus distintas facetas. Tanto para análisis de capacidades como para evaluar diseños de nuevas instalaciones y escenarios de tráfico o parámetros distintos a los observados normalmente.

4.2. Descripción del problema

4.2.1. El aeropuerto

El caso al que vamos a dedicar este trabajo está centrado, tal como se dijo en la sección 1, en el Aeropuerto Internacional de Carrasco. Esta aeroestación se encuentra en un proceso de renovación por el cual grandes mejoras se van a ver reflejadas una vez que se termine. Entre ellas encontramos la

6. Janic, M.: "Terminal sizing theory", Loughborough University (2000)

7. Joustra, P. E., Van Dijk, N. M.: "Simulation of check-in at airports", Proceedings of the 2001 Winter Simulation Conference

8. Gatersleben, M. R., van der Weij, S. W.: "Analysis and simulation of passenger flows in an airport terminal", Proceedings of the 1999 Winter Simulation Conference

9. Verbraeck, A., Valentin, E.: "Simulation building blocks for airport terminal modeling", Proceedings of the 2002 Winter Simulation Conference

10. Kiran, A. S., Cetinkaya, T., Og, S.: "Simulation modeling and análisis of a new international terminal", Proceedings of the 2000 Winter Simulation Conference

11. Takakuwa, S., Oyama, T.: "Simulation análisis of international-departure passenger flows in an airport terminal", Proceedings of the 2003 Winter Simulation Conference

12. Pendergraft, D. R., Robertson, C. V., Shrader, S.: "Simulation of an airport passenger security system", Proceedings of the 2004 Winter Simulation Conference

prolongación de su pista principal, la construcción de nuevas calles de rodaje para aeronaves, un nuevo sistema de iluminación para las pistas, una nueva plataforma de estacionamiento de aeronaves, una nueva terminal de cargas, y como más importante y objeto de nuestro estudio, una nueva terminal de pasajeros.

Este nuevo edificio de pasajeros contará con la última tecnología en cuanto a funcionalidad y diseño del mismo. Será lo que IATA (2004) define como una terminal centralizada, lineal, de dos niveles y medio. O sea, que como estamos hablando de un aeropuerto de dimensiones pequeñas y de un tráfico bastante homogéneo, podemos pensar en un modelo centralizado, que implica que tendremos una única terminal con subsistemas de proceso de pasajeros unificados en cada una de sus etapas. En contraposición, encontramos aeropuertos descentralizados, donde existen distintas terminales de proceso de pasajeros en el lado tierra y distintos accesos a la zona operativa de aeronaves en el lado aire. Esto se da en casos de mucho tráfico o de topologías de pasajeros bien diferenciadas donde es útil y eficiente contar con más de una terminal y más de un edificio o espigón de acceso al lado aire. Por ejemplo, terminales distintas para pasajeros internacionales y domésticos y edificios distintos de acceso a aviones en función del tamaño de los mismos. Se la denomina lineal, porque su forma de acceso al lado aire además de ser centralizado está configurado de forma que todas las aeronaves se conectan con el edificio estacionadas sobre la plataforma alineadas entre sí. Es decir una al lado de otra de frente a la terminal, y la forma de crecimiento que tiene el edificio es hacia los costados agrandando la línea sobre la cual estacionan los aviones. Para entender el concepto, podemos nombrar otros como la de muelles que emulan una marina de estacionamiento de barcos, con una pasarela central y aviones de ambos lados, y allí el crecimiento se da agregando muelles. También existe el concepto de satélites o el modular, para más información consultar IATA (2004 p.301). Los niveles del edificio son muy importantes, ya que son los que dan una medida de organización y confort más importante a un aeropuerto. Uno de dos pisos, o niveles, tiene separados los flujos de arribos y partidas de manera total, el nivel inferior, generalmente alineado con el terreno, es el que tiene todos los servicios y flujos de pasajeros que arriban a la terminal desde los aviones. Y el nivel superior es el que maneja los flujos de pasajeros que llegan al edificio para tomar un vuelo. Es decir, cada nivel tiene un acceso vehicular, que en el caso del superior sirve para que los pasajeros desciendan y entren a la terminal para realizar en ese nivel todos los procesos necesarios para acceder al avión a través de una manga, sin necesidad de usar escaleras o ascensores. Lo mismo pasa con los flujos de arribos que entran a través de las mangas y hacen uso de un nivel intermedio, un entrepiso que reúne todas las mangas y lleva a los pasajeros en conexión a un puesto de chequeo que les permite acceder al piso superior para llegar a su nuevo vuelo, y a los pasajeros que terminan su viaje, al nivel inferior para hacer controles de migraciones, retirar su equipaje y pasar por aduana para salir del edificio.

En la figura 9 vemos una imagen de la futura terminal de Carrasco, en ella podemos observar las características del edificio en sus dos niveles con los accesos vehiculares para cada nivel.



Figura 9 Imagen generada por computadora de la nueva terminal

Los aeropuertos y la industria aeronáutica en general tienen varias organizaciones internacionales, al margen de las que pertenecen a los gobiernos de cada país y regulan la actividad local, que se dedican a nuclear sectores de la industria para compartir experiencias, mejorar la actividad y generar políticas y recomendaciones sobre diversos aspectos. Entre las más importantes encontramos a OACI (Organización de Aviación Civil Internacional), que nuclea en mayor medida a los organismos gubernamentales que regulan la actividad y por ende su mayor foco pasa por la seguridad y todo lo relacionado con el área operativa de los aeropuertos; el ACI (Airports Council International), que nuclea a los operadores aeroportuarios y que recién hace algunos años tomó relevancia internacional debido a las privatizaciones llevadas a cabo a nivel mundial que separaron a los operadores aeroportuarios de los gobiernos; y a IATA (International Air Transport Association) que nuclea principalmente a las aerolíneas y que desde hace mucho tiempo genera estándares y recomendaciones para toda la industria a nivel mundial. Esta última proporciona los índices de calidad y confort que usaremos en nuestro trabajo.

La construcción de la nueva terminal de pasajeros del Aeropuerto Internacional de Carrasco, es una de las tantas exigencias de la concesión efectuada por el gobierno uruguayo a un operador privado. Dentro de éstas, encontramos innumerables detalles de todo tipo relativos a la construcción misma y a tecnologías y normas a aplicar. En cuanto a los niveles de confort a los pasajeros, relacionados con los espacios del edificio, se decidió aplicar las recomendaciones IATA como exigencia de diseño. Así como se explicó en la sección 1 de este trabajo, existen categorías asignadas a los aeropuertos según lo que IATA llama LoS (Level of Service o Nivel de Servicio) en las distintas áreas destinadas al flujo de pasajeros. Estas van de A a E y son descritas de la siguiente manera:

- A: Un nivel de servicio excelente. Condiciones de flujo libre, sin demoras y excelentes niveles de confort.
- B: Alto nivel de servicio. Condiciones de flujo estables, muy pocas demoras y altos niveles de confort.
- C: Buen nivel de servicio. Condiciones de flujo estables, demoras aceptables y buen nivel de confort.
- D: Nivel de servicio adecuado. Condiciones de flujo inestables, niveles de demora aceptables por períodos cortos de tiempo y niveles de confort adecuados.
- E: Nivel de servicio inadecuado. Condiciones de flujo inestables, demoras inaceptables y niveles de confort inadecuados.
- F: Nivel de servicio inaceptable. Condiciones de flujos cruzados, roturas del sistema y demoras inaceptables. Niveles de confort inaceptables. Este nivel generalmente no se tiene en cuenta por ser completamente inaceptable.

En el siguiente cuadro vemos como se clasifica cada una de las áreas de un aeropuerto según sus espacios disponibles.

Área	A	B	C	D	E	F
Colas de check-in	1,8	1,6	1,4	1,2	1,0	Rotura del Sistema
Circulación – Espera	2,7	2,3	1,9	1,5	1,0	
Área de embarque	1,4	1,2	1,0	0,8	0,6	
Cintas de equipaje	2,0	1,8	1,6	1,4	1,2	
Migraciones – Seguridad	1,4	1,2	1,0	0,8	0,6	

Figura 10 Tabla de Niveles de Servicio IATA. (metro cuadrado/ocupante)

De acuerdo al diseño del Aeropuerto de Carrasco, las áreas destinadas al flujo de pasajeros no deberían ser de una categoría inferior a C. En este trabajo vamos a realizar la simulación para ver en qué categoría podemos encuadrar a cada una de las áreas del subsistema de partidas.

Es importante tener en cuenta algunos datos relevantes sobre el aeropuerto que se detallan en el cuadro de la figura 11.

Pasajeros anuales 2007	1,1 millones
Pasajeros arribando en hora pico	300
Superficie total terminal	49.000 metros cuadrados
Superficie área check-in	1561 metros cuadrados
Superficie área seguridad	376 metros cuadrados
Superficie área migraciones	552 metros cuadrados

Figura 11 Datos relevantes del aeropuerto

4.2.2. Los procesos del subsistema de partidas

En la mayor parte de los aeropuertos existen una serie de subsistemas comunes que son en esencia los mismos procesos, pero que en cada uno presentan particularidades vinculadas a la legislación local y la topología de la composición de pasajeros y destinos. En este trabajo nos vamos a centrar en el sistema de partidas del aeropuerto. Este cubre desde que los pasajeros llegan al aeropuerto hasta que se suben al avión, pasando por varios procesos intermedios.

En primer lugar y como elemento común a todos los aeropuertos encontramos el proceso de check-in. En este trámite el pasajero se presenta a un mostrador de la línea aérea correspondiente para registrarse como tal y entregar el equipaje que irá a la bodega del avión. Dependiendo del destino y la línea aérea este proceso consiste en una serie de pasos que puede ser mayor o menor. Siempre se debe buscar al pasajero entre las reservas que existen para su vuelo, y una vez identificado, vincularlo con las piezas de equipaje que desea entregar, luego se debe asignar un asiento y emitir el boarding pass o documento de embarque junto con los tickets que identifican a las maletas que viajan junto a él. Además de estos pasos estándar, se pueden tener en cuenta procesos que tienen que ver con normas de seguridad como cuestionarios obligatorios o un chequeo secreto por parte del personal de la aerolínea por el cual se marca al pasajero como seleccionado para ser revisado exhaustivamente en el proceso de seguridad posterior. También se pueden dar procesos comerciales como la venta de un pase de una clase a otra o el cobro de sobrepeso si es que el equipaje supera en peso lo establecido por la aerolínea. Todo este proceso se da en mostradores que tienen una cantidad finita instalada en el aeropuerto. Y que además son asignados a cada aerolínea en función de la cantidad de vuelos que tenga. En otros tiempos, cada aerolínea tenía su sector asignado dentro de los aeropuertos, ya que los mostradores tenían terminales del sistema específico de la aerolínea y no podían ser usados por otra. Hoy en día, la mayoría de los aeropuertos cuenta con lo que se conoce como C.U.T.E. (siglas en inglés de Common Use Terminal Equipment), que consiste en terminales estándar con un software que permite a cualquier usuario de cualquier aerolínea acceder a su sistema. Esto permite un uso muchísimo más eficiente de los recursos, ya que en el sistema anterior si una aerolínea sólo tenía dos vuelos diarios a las 10 AM y a las 10 PM, el o los mostradores asignados permanecían cerrados la gran parte del día y ese recurso se mal utilizaba. Gracias al sistema C.U.T.E. los mostradores pueden asignarse dinámicamente a cada aerolínea según sus necesidades de uso en cada día. Otro aspecto importante a tener en cuenta a la hora de la simulación de las colas de una operación de check-in es la forma en que ellas se organizan. Para esto existen dos formas usuales, teniendo una cola por cada mostrador o una gran cola que da acceso a un grupo de mostradores que tienen la misma función (una misma aerolínea o un vuelo específico). En nuestro caso implementamos la segunda opción con lo que se denomina banco de recursos (resource banking) teniendo una única cola para llegar a ellos y dividir equitativamente las cargas de trabajo. En el aeropuerto de Carrasco esto se da por aerolínea. Respecto a los tiempos de proceso, dependiendo de cada aerolínea y tipo de pasajero, existen tiempos medios, mínimos y máximos. En nuestro caso tomaremos un promedio generado por la autoridad aeronáutica uruguaya y el departamento de infraestructura del aeropuerto, éste se verá reflejado en las especificaciones del modelo a generar en la próxima sección.

Siguiente al proceso de check-in, en nuestro modelo está el chequeo de seguridad por parte de la Policía Aeronáutica. Recientemente OACI ha emitido una nueva recomendación respecto a este proceso, actualizándolo luego de amenazas al tráfico aéreo realizadas con explosivos líquidos. Esto hace aún más exhaustivo el proceso, que ya desde los episodios de septiembre de 2001 se había transformado en algo muchísimo más complejo que en los años previos a estos eventos. Este proceso, en nuestro caso, cuenta con dos pasos, el primero obligatorio y el segundo opcional. Todos los pasajeros deben pasar por un arco detector de metales y pasar su equipaje de mano por una máquina de rayos x. Luego, sólo aquellas piezas de equipaje que resultan sospechosas para quien controla la máquina de rayos x son chequeadas en una segunda etapa por un oficial que la revisa a mano.

A continuación, tenemos el proceso de migraciones, donde se controla la documentación de cada pasajero para salir del país. Está claro que este paso es sólo para salir del país, pero nuestro aeropuerto modelo tiene un 98,5% de pasajeros internacionales, por lo cual vamos a despreocupar los de cabotaje y considerar obligatorio el proceso de migraciones. Este proceso es muy parecido para todos los pasajeros, salvo para aquellos que viajan con documentación que cuente con la codificación OACI que permite leer con un scanner muchos datos que, de otra forma, el operador debe tipear.

La frontera de nuestro modelo está en este último proceso. Es decir, que por más que después de esto los pasajeros todavía deben realizar otras operaciones antes de dejar el aeropuerto hacia el avión, nosotros no vamos a considerar nada más en nuestro modelo.

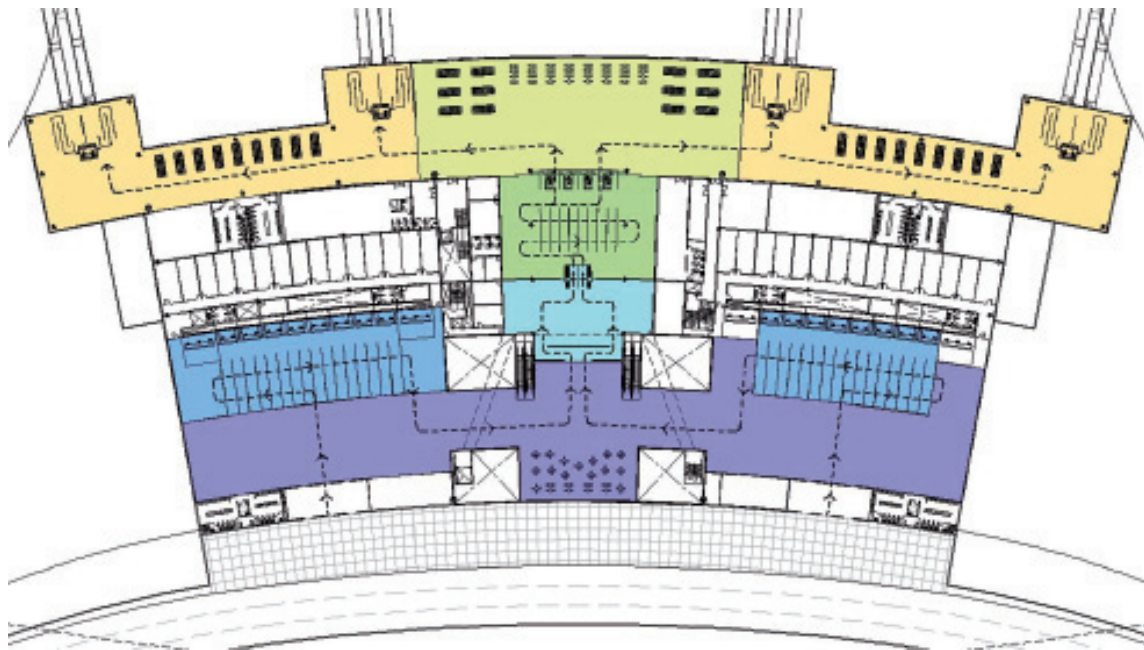


Figura 12 Diagrama de flujo de pasajeros de partidas

En la figura 12 vemos un diagrama de flujo de pasajeros de partidas sobre un plano del Aeropuerto Internacional de Carrasco. En la parte inferior encontramos la vialidad que lleva a los vehículos hasta la Terminal y por donde acceden los peatones al edificio. De allí tenemos el recorrido marcado con flechas y líneas punteadas. Primero, la zona celeste es el primer paso donde el pasajero hace la cola de check-in, luego la sección de color más claro, donde se hace la cola de seguridad y por último el área verde, donde se realiza la espera para el proceso de migraciones.

4.3. Modelo del Aeropuerto de Carrasco

En esta sección desarrollaremos el modelo específico del aeropuerto en Arena para realizar su análisis mediante la simulación. También usaremos esta sección como especificación de requerimientos del modelo como lo propone Kelton et al (2007) para proyectos en los cuales un mismo individuo o equipo es quien requiere el estudio y quien lo realiza a la vez.

4.3.1. Objetivos y métricas del modelo

Como se planteó en la sección 1 el objetivo de esta simulación es predecir la categoría IATA de nivel de servicio en que se encontrará la nueva terminal del aeropuerto de Carrasco que se encuentra en construcción. Para esto debemos usar medidas concretas que, luego realizada la simulación, nos lleven a una conclusión sin ambigüedad a partir de sus resultados.

En este caso la medida que vamos a calcular es la cantidad de ocupantes por metro cuadrado en el momento de mayor ocupación de esa área durante la simulación. Para eso tomaremos el valor máximo de la o las colas del modelo por cada proceso y lo dividiremos por la superficie del área correspondiente. Es decir, no llegaremos a una única categoría sino a una categoría para cada área de medición. Para esto usaremos en cada caso un dato proveniente de la simulación, y otro que extraeremos de la tabla de la figura 11. Una vez obtenido el cociente entre ocupantes máximos y superficie del área, lo compararemos con los presentados en la tabla de la figura 10 para determinar la categoría de cada área.

Las medidas a tener en cuenta como resultado del trabajo son las siguientes:

Número máximo en colas de check-in / superficie área check-in

Número máximo en colas de seguridad / superficie área seguridad

Número máximo en colas de migraciones / superficie área migraciones

Estos tres cálculos nos proporcionarán el número máximo de ocupantes por metro cuadrado en cada área.

De esta manera, con estos datos podremos construir la información necesaria para llegar a la conclusión requerida que tendrá la forma de una categoría entre la A y la E por cada una de las áreas de proceso estudiadas.

4.3.2. Límites específicos del modelo

En la sección 1.3. del trabajo se expusieron los límites generales a la producción del trabajo en términos de cada una de la secciones principales del mismo. A partir de esto, y de la descripción de los procesos que se llevan a cabo en el sistema real, detallada en la sección 4.2.2., vamos a definir los límites y la frontera específica del modelo respecto al sistema real.

La mayor limitante de este trabajo respecto al sistema real será dada por la **versión de Arena** que estamos utilizando. Ésta está **destinada a uso académico** y por tanto no permite la generación de modelos de tamaño importante. Es por eso que **limita la cantidad de entidades** que pueden estar en el modelo de manera concurrente a **150**. Este número nos permite generar el modelo y hacer pruebas sobre su validez, pero no nos va a permitir hacer pruebas de estrés que nos muestren los límites operativos del sistema. Y para nuestro caso, nos va a obligar a bajar la cantidad de pasajeros reales para poder correr el modelo. Esto nos va a permitir analizar la lógica y los cuellos de botella del sistema, así como la lógica para el cálculo de la categoría IATA del aeropuerto, que será incluida en el sistema, pero no nos permitirá hacerlo con el número de pasajeros necesario para llegar a una conclusión veraz.

El hecho de que estamos generando una representación abstracta del sistema implica que se harán simplificaciones del mismo a efectos prácticos, siempre teniendo en cuenta que éstas no sean causa de comportamientos que alteren la performance del modelo versus el sistema y específicamente de los parámetros a analizar.

En este caso, a efectos de simplificar la generación del modelo se han hecho algunos compromisos respecto a ciertos parámetros del sistema. En primer lugar, la generación de pasajeros podría dividirse en distintos criterios e identificar a las entidades para realizar diferencias en los procesos del sistema. Por ejemplo, podríamos tomar pasajeros de clase ejecutiva o económica, domésticos o internacionales. Nuestro modelo no requiere generar estas divisiones para medir los parámetros buscados, es por eso que sólo tendremos una entidad tipo pasajero. El proceso de check-in también puede tener distintas modalidades. Existen en muchos aeropuertos del mundo kioscos de check-in automatizados que permiten a los pasajeros realizar el proceso sin intervención de personal de la aerolínea. También existen distintas políticas por aerolínea según el tipo de pasajero o destino en la realización del proceso. En nuestro caso tomaremos un criterio uniforme respecto al proceso y los tiempos para todas las operaciones. También usaremos un criterio uniforme para definir la cantidad de recursos de check-in operativos. Es decir, cada aerolínea tiene políticas distintas respecto la cantidad de mostradores a utilizar para despachar un vuelo, en nuestro modelo no vamos a tener eso en cuenta y en cambio, vamos a utilizar la recomendación IATA que dice que para despachar vuelos que utilizan aviones wide-body (de porte grande, ver glosario) se requieren 6 mostradores y para despachar vuelos que utilizan aviones narrow-body (de porte mediano y pequeño, ver glosario) se requieren 3 mostradores. En algunos aeropuertos, incluyendo el de Carrasco, existe un proceso intermedio entre el check-in y el proceso de seguridad, el de pago de tasas aeroportuarias. En nuestro modelo contemplamos la eliminación de este paso.

Además de los procesos que pueden modelarse de manera simplificada, tenemos un aspecto clave del modelo que resultaría imposible de modelar de forma precisa. Esto es el comportamiento humano. Es por eso que el modelo toma ciertas consideraciones respecto al comportamiento que hoy resulta muy complejo de definir. En primer lugar, sólo vamos a considerar a los pasajeros en el modelo, no consideraremos a los acompañantes que puedan llegar con ellos al aeropuerto y que podrían ser usados para tomar otras medidas de confort y operatividad. En segundo lugar, el comportamiento de los pasajeros es modelado de tal forma que no se considera la posibilidad de otras actividades fuera de la realización de los procesos requeridos para embarcar, como ser detenerse en los negocios o locales de gastronomía, etc. Tampoco estamos considerando la velocidad de caminata de los pasajeros entre procesos, ya que consideramos que este tiempo no tiene injerencia en los resultados buscados por el experimento de simulación. Dentro de esta categoría de comportamientos humanos, consideramos que los recursos que deben ser operados por el gobierno, como los mostradores de migraciones y el control de seguridad

siempre tienen el personal necesario para su operación. Ésto último no es siempre cierto en el sistema real, pero a modo del análisis que estamos llevando a cabo tiene sentido tomar el escenario ideal.

Tomando todas estas suposiciones veremos en la próxima sección cómo se genera el modelo dentro de la herramienta Arena.

4.3.3. Modelo en Arena

Tal como se explicó en la descripción de los procesos del sistema a estudiar de la sección 4.2.2., el primer proceso de este subsistema es el de check-in de los pasajeros, sin embargo para poder realizar nuestra simulación debemos comenzar por un aspecto clave antes de poder modelar el proceso de check-in.

En este modelo las entidades serán de un único tipo y éstas representarán a los pasajeros. En la figura 13 vemos la lista de las entidades en Arena y la configuración de la entidad pasajero donde definimos el nombre de la entidad, su figura para la animación y la marcamos para generar estadísticas sobre la misma.

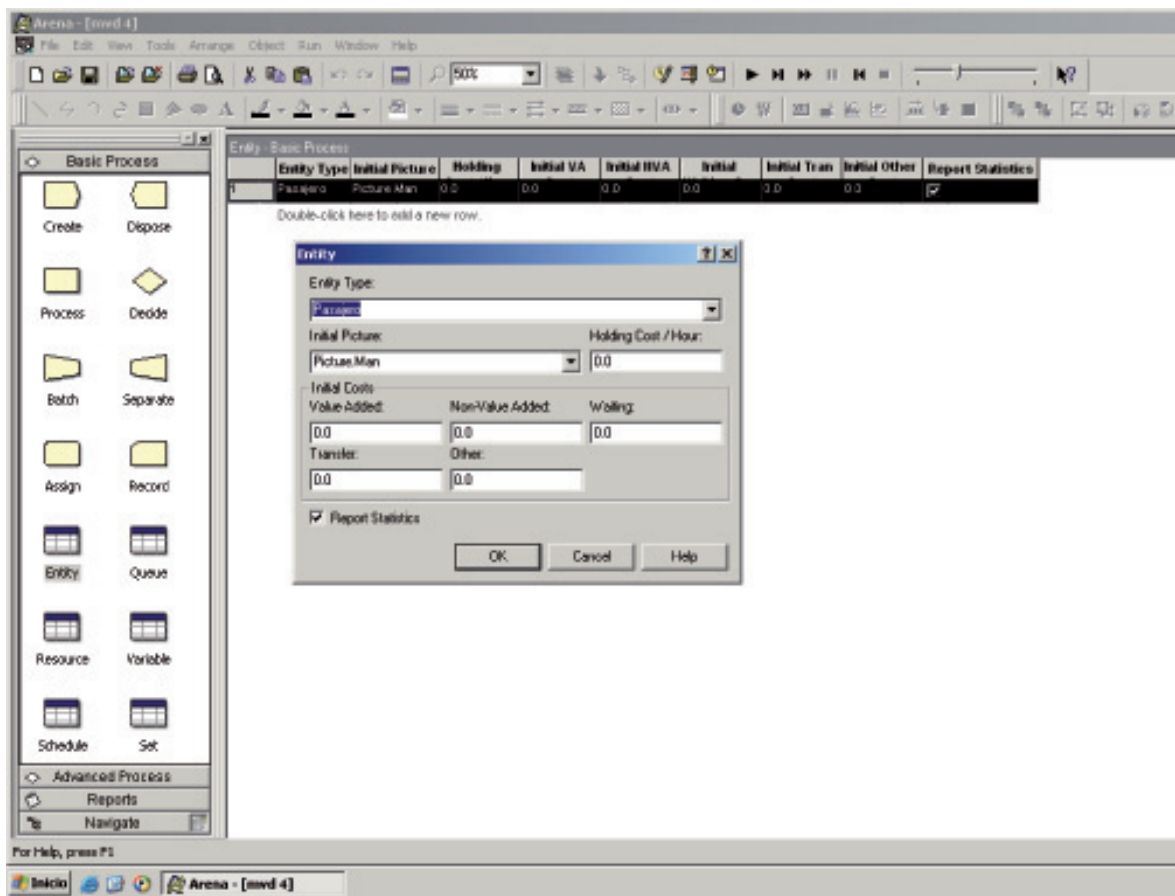


Figura 13 La configuración de las entidades de nuestro modelo en Arena

Posiblemente, la parte más importante del modelo sea cómo generar las entidades para que la llegada de pasajeros al aeropuerto simulado sea representativa de cómo se da en el sistema real. En nuestro modelo, hemos tomado una serie de criterios que responden a observaciones del sistema real y a experiencias de trabajos similares que se han recopilado para la realización de este trabajo. Varias de estas experiencias están citadas en la sección 4.1. del presente. En primer lugar, se observa que por la programación de los vuelos del aeropuerto estudiado, podemos tomar cada día por separado para su análisis, ya que durante la madrugada no existen operaciones dejando a cada jornada de operación separada, sin solapamiento ni intersección de ningún tipo con las jornadas previas o posteriores. Dado esto, hemos decidido hacer la simulación de un día pico y así realizar la categorización IATA, tomando en cuenta la mayor afluencia de pasajeros observados en el sistema real. De otra manera perdería sentido el estudio, ya que nuestra conclusión podría ser equivocada porque estamos teniendo en cuenta cantidades de pasajeros bajas y nuestro estudio requiere de picos de ocupación para poder ser efectivo. Recurriendo a planillas de programación del aeropuerto encontramos que el día pico de la semana es el jueves, tanto por la cantidad de vuelos como por la de pasajeros. En la figura 14 vemos una planilla con la información de los vuelos programados.

MATRÍCULA DE LA AERONAVE		MARCA Y MODELO	EMPRESA	Nº DE VUELO	PROC. / DEST.	POS.	CIENSA	PARTIDA	Capacidad APT	Load Factor	Total Pax
								PROG.	Pax	Promedio	
	B-732	PUA	222 / 7361	GRU			06:05	110	70.00%	77	
	B-738	GLO	7487	POVGRU			06:10	189	75.00%	142	
	B-732	PUA	151	AEP			07:00	110	65.00%	72	
	B-732	ARG	1203	AEP			08:25	110	65.00%	72	
	B-733	ARG	1207	AEP			09:45	120	65.00%	78	
	B-732	PUA	155	AEP			10:00	110	65.00%	72	
	B-752	PUA	220 / 7367	GIG			10:00	200	70.00%	140	
	F-100	LAP	702 / 7707	EZEIASU			10:10	90	80.00%	72	
	SF-34	BR	5731	AEP			10:30	34	50.00%	17	
	B-732	ARG	1219	AEP			11:00	110	45.00%	50	
	B-732	PUA	401	SCL			11:15	110	70.00%	77	
	A-320	LAN	901	SCL			11:30	154	75.00%	116	
	B-732	PUA	159	AEP			12:30	110	45.00%	50	
	B-733	ARG	1205	AEP			12:40	120	50.00%	60	
	A-343	IBE	6864	MAD			14:25	295	80.00%	238	
	B-752	PUA	230 / 7363	GRU			15:20	200	70.00%	140	
	B-732	LAN	7532	ROS/SCL			17:00	110	65.00%	72	
	B-738	GLO	7489	POVGRU			17:05	189	75.00%	142	
	SF-34	BR	5737	AEP			17:15	34	65.00%	22	
	B-732	ARG	1223	EZE			17:20	110	60.00%	66	
	B-732	PUA	167	AEP			17:30	110	65.00%	72	
	A-320	LAN	903	SCL			17:45	154	75.00%	116	
	B-763	AAL	900	EZE/IMA			17:50	269	40.00%	108	
	B-732	ARG	1209	AEP			18:00	110	70.00%	77	
	B-763	UAL	846	EZE/WAS			18:25	269	35.00%	94	
	B-732	PUA	171	AEP			20:00	110	75.00%	83	
	B-732	ARG	1213	AEP			20:20	110	70.00%	77	

La distribución de la presente es reservada para las autoridades designadas en el A.I.P. / La información aquí contenida no es oficial y está sujeta a cambios.

Figura 14 Planilla de programación de operaciones diaria

En segundo lugar, vamos a tomar de la planilla de operaciones de la figura 14 las cantidades de pasajeros para modelar su arribo. Este cálculo aproximado de pasajeros proviene de otros datos encontrados en la planilla. Dentro de ésta encontramos una lista donde cada fila representa un vuelo y tiene sus características en cada columna. Es decir, en la segunda columna de la izquierda de la misma, encontramos el título "Marca y Modelo", allí encontraremos la descripción de la aeronave usada para dicho vuelo. Luego encontramos los datos sobre la aerolínea que realiza la operación, el número de vuelo, el destino, y la hora de partida. En las siguientes columnas encontramos la información necesaria para calcular la cantidad de pasajeros esperada, que se refleja en la última columna sobre la derecha de la planilla (pax significa pasajero en la jerga aeroportuaria y del transporte en general). Esta información es la capacidad de la aeronave a utilizar, definida por su marca y modelo (por ejemplo, un Boeing 737-200 o B-732 tiene una capacidad de 110 pasajeros), y el factor de ocupación (Load Factor) promedio histórico de dicho vuelo. Es decir, si el vuelo de Pluna (PUA es el código IATA de la aerolínea) número 222 con destino a San Pablo (GRU es el código IATA del aeropuerto de Guarulhos en San Pablo) sale todos los días en una aeronave con un promedio de ocupación del 70% en una aeronave que puede llevar 110 pasajeros, el número de

pasajeros esperados es de 77 (70 % de 110). Hay que tener en cuenta que el promedio de ocupación se calcula de manera mensual teniendo en cuenta cada una de las operaciones de un vuelo.

A partir de los horarios de partida de los vuelos y de la cantidad de pasajeros esperada para cada vuelo es que vamos a modelar el arribo de los pasajeros de cada vuelo. Métodos similares al elegido por nosotros son descritos en Joustra y Van Dijk (2001) y en Pendergraft et al (2004), donde basan el patrón de llegada de pasajeros al aeropuerto en una serie de parámetros cuantitativos y cualitativos que tienen como valor clave la hora de salida de los vuelos programados. De la misma manera que en los trabajos recién mencionados, vamos a modelar el arribo de pasajeros al aeropuerto a través de distribuciones probabilísticas de tipo Poisson no estacionarias. Esta distribución nos permite definir rangos de tiempo y asignarle a éstos una tasa de arribos. Es decir, para nuestro proyecto, podemos definir franjas de una hora a lo largo del día y asignar una tasa de arribo de pasajeros por cada hora. Por ejemplo, podemos decir que entre las 13 y las 15 hs. llegan al aeropuerto 100 pasajeros por hora. Dentro de cada franja horaria se generan las entidades con una distribución exponencial con una media en la tasa fijada, en nuestro caso 100. De esta manera se obtiene el comportamiento aleatorio a la vez que se controla el patrón de arribo de pasajeros.

De esta manera, podemos tomar parámetros estándar fijados por IATA para modelar las tasas de arribo de pasajeros hora por hora. Éstos nos indican que los mostradores de check-in abren tres horas antes de un vuelo y cierran media hora antes de la partida del mismo. Con lo cual podemos concluir que, la cantidad de pasajeros para un vuelo llegará, a la cola del check-in desde tres horas antes, hasta media hora antes del mismo, y que según observaciones de las aerolíneas podemos concentrar un 70% de los arribos entre las dos horas previas a la partida. Es decir, si un vuelo sale a las 8 de la mañana podemos concentrar al 70% de los arribos de pasajeros a la terminal entre las 6 y las 7 de la mañana.

Usando estas premisas y la información de la figura 14 podemos generar un modelo preciso de los arribos de pasajeros al Aeropuerto de Carrasco para los días jueves. O sea que podríamos volcar a una distribución Poisson no estacionaria, las tasas arribo de pasajeros hora por hora del día jueves tomando en cuenta la cantidad de pasajeros de cada vuelo de la planilla, distribuyendo los arribos desde tres horas antes de la partida del avión y concentrando la mayoría de los arribos entre la hora previa y la anterior a la salida del vuelo. En este caso vamos optar por añadir una variante a éste, en lugar de generar una sola distribución diaria, vamos a dividirla y generar una para cada aerolínea del aeropuerto.

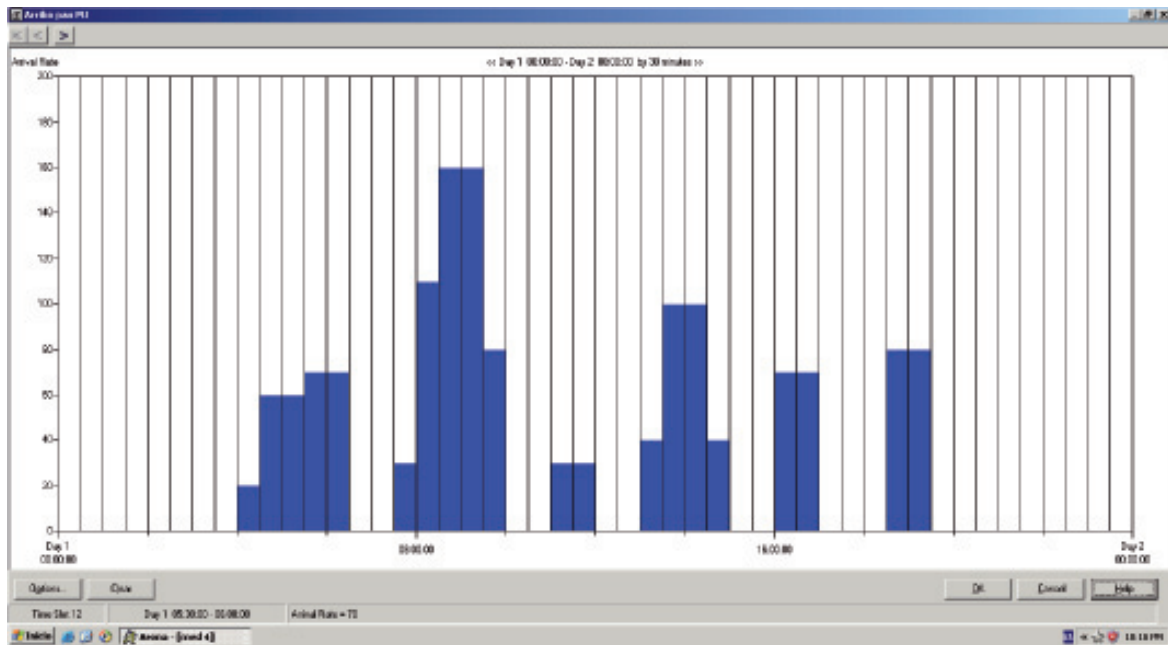


Figura 15 Distribución de tasas de arribo de pasajeros para Pluna

Como mencionamos en la sección 3.2.1. el módulo Create de Arena es el encargado de la generación de entidades en un modelo. En nuestro caso, hemos decidido insertar uno de estos módulos por cada aerolínea que opera en el aeropuerto por motivos relacionados con el próximo proceso, el de check-in. Cada uno de estos módulos fue configurado de tal manera que genere los arribos de pasajeros en base a una distribución Poisson no estacionaria creada conforme a las premisas expuestas anteriormente. En la figura 15 vemos la distribución creada para Pluna, la aerolínea con mayor cantidad de vuelos y pasajeros

de nuestro modelo. Allí vemos en el eje y las tasas de arribo de pasajeros por hora y en el x las horas del día que han sido divididas en secciones de 30 minutos. Para crear estas distribuciones tenemos que recurrir al módulo Schedule del Basic Process Panel y luego vincular la instancia del módulo creado con el Create para que éste lo use para generar entidades. En nuestro caso el módulo Schedule de la figura 15 se llama Arribo Pax PU y en la figura 16 podemos ver como configuramos el módulo Create llamado Arribo de pax a la terminal Pluna para que utilice el Schedule Arribo pax PU.

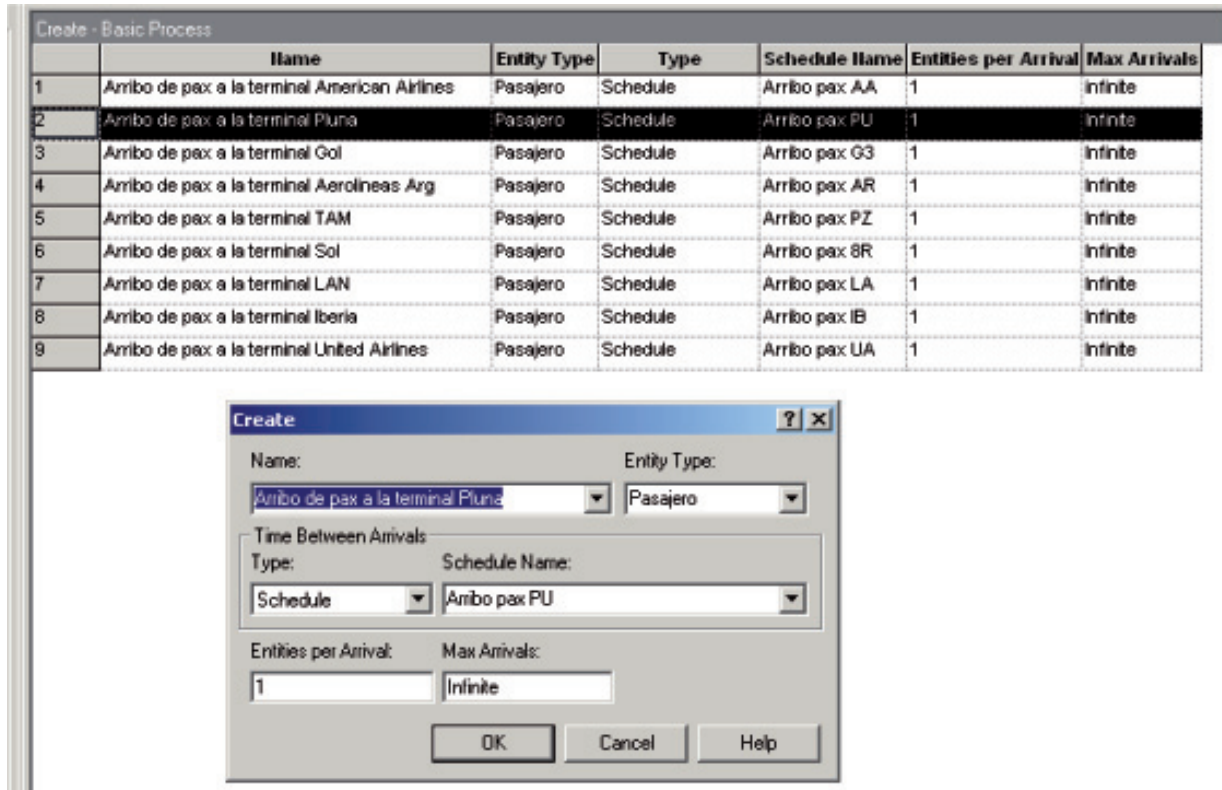


Figura 16 Configuración del módulo Create para arribo de pasajeros de Pluna

También podemos ver en la figura 16 el listado de todos los módulos Create de nuestro modelo y sus configuraciones. Para tener una idea completa de cómo es el modelo construido del sistema de partidas del Aeropuerto de Carrasco, en la figura 17 se encuentra el diagrama de flujos generado en Arena. En el Anexo B existe una impresión en mayor tamaño del mismo.

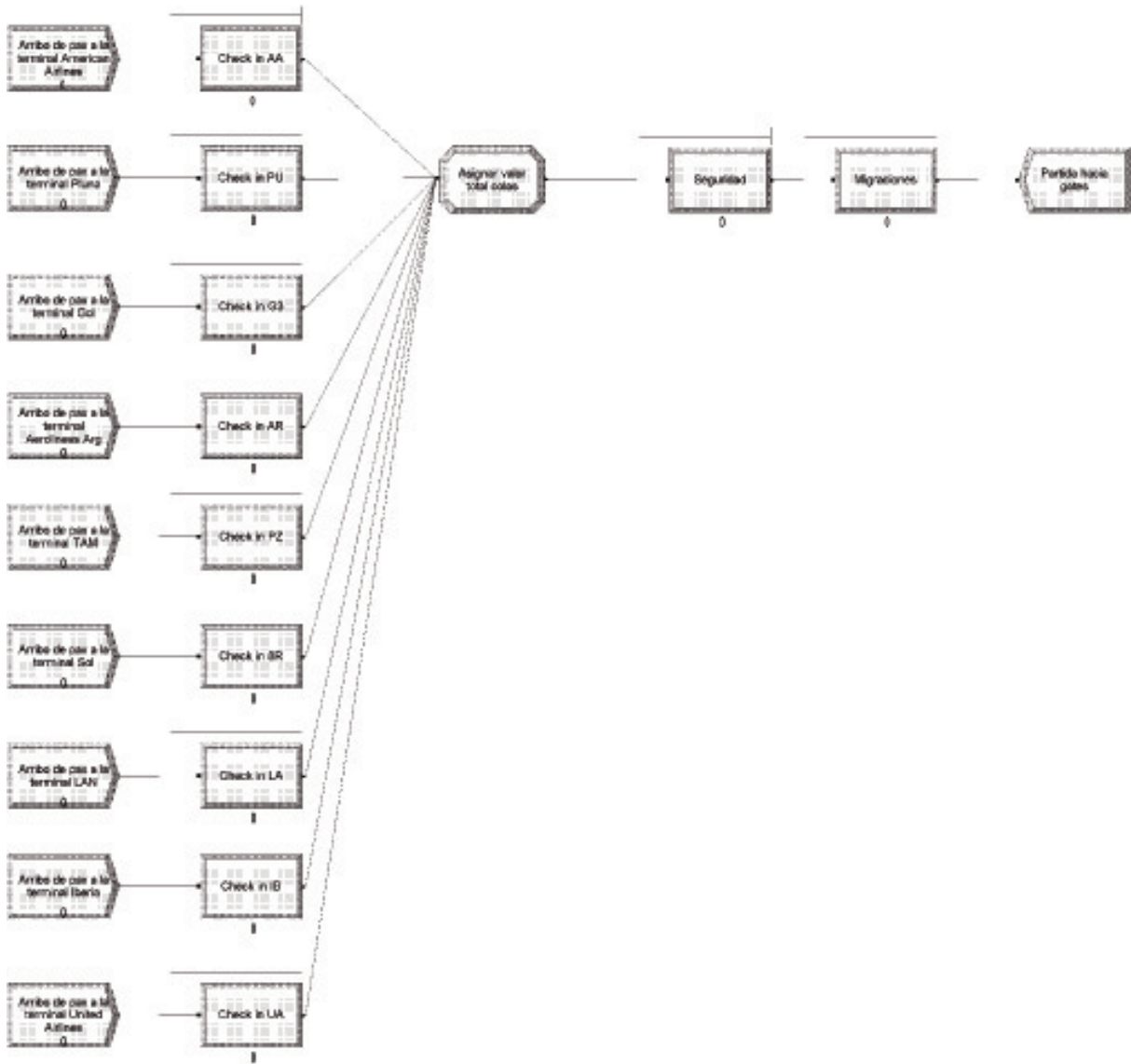


Figura 17 Diagrama de flujos completo de nuestro modelo

Ya hemos visto como están configurados los módulos Create junto con la lógica de generación de entidades. Ahora veremos como funciona el proceso de check-in.

En la sección 4.2.2. hemos visto las opciones de configuración de colas posibles en un aeropuerto, y hemos señalado la que se usa en nuestro aeropuerto y por ende en nuestro modelo. Esta configuración es la que agrupa varios mostradores y forma una sola cola de pasajeros que van accediendo a cualquier mostrador que esté disponible. En nuestro caso la agrupación de mostradores se hace por aerolínea, es decir, que sin importar los vuelos, destinos o clases de pasajeros, todos los mostradores de una aerolínea tienen una cola común y todo aquel que desea ser chequeado en un vuelo de la misma deberá usarla. Debido a esto es que hemos modelado la generación de pasajeros dividida por aerolínea, ya que cada módulo Create genera pasajeros que entran directamente en la cola creada para los mostradores de check-in de su respectiva aerolínea.

Dada esta configuración, es que vemos en el modelo de la figura 17 que cada módulo Create está conectado con su respectivo módulo de proceso de check-in. Lo que vemos en la figura 17 dibujado sobre cada módulo de check-in es la representación de la cola por la que pasan las entidades antes de hacer uso de los servidores ubicados en cada mostrador para realizar el proceso.

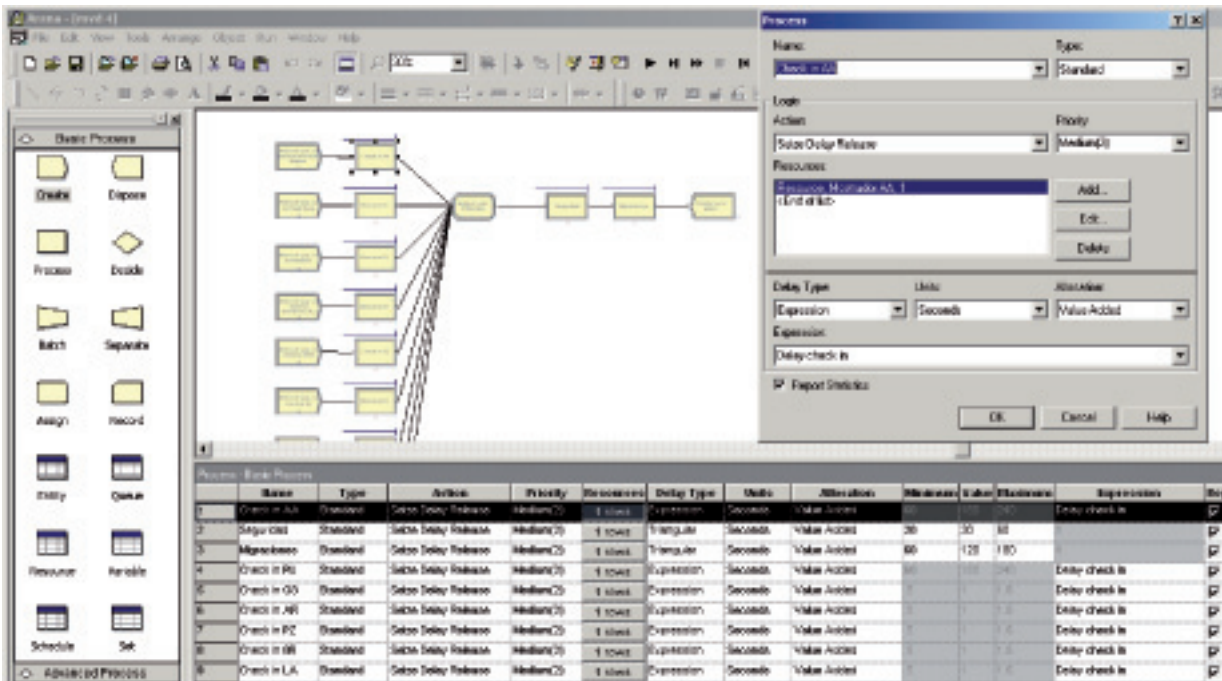


Figura 18 Configuración de módulo Process usado para modelar el check-in

El módulo Process, que se ve en la figura 18, es el que sirve para modelar procesos de todo tipo y generan un “retraso” en el camino de las entidades a través del diagrama de flujos. Es decir, que para representar un proceso realizado sobre la entidad, la simulación detiene a la misma en ese módulo imitando el tiempo que llevaría en el sistema real la ejecución de la operación simulada. Para esto Arena prevé que se utilice o no un recurso. Los recursos de un modelo fueron presentados en la sección 2.1.2. y son aquellos que dan servicio a las entidades, en este caso mostradores de check-in. El proceso requiere un recurso y por eso en el campo Action (figura 18) elegimos la opción Seize Delay Release, que significa tomar el recurso, generar el retraso, y soltar el recurso. O sea que mientras la entidad esta realizando el proceso toma de forma exclusiva el recurso, impidiendo que este sirva a ninguna otra entidad hasta que, luego de transcurrido el tiempo de proceso, se libera y queda disponible para atender a otra entidad. Tal como lo mencionamos, el siguiente parámetro de importancia en la configuración de módulo Process que representa el check-in, es la definición de los recursos a utilizar. Para eso, como vemos en la figura 18, Arena permite generar una lista de recursos a utilizar en ese módulo. Esta lista debe llenarse con recursos que pueden ser elegidos sólo si previamente fueron definidos en el módulo de datos Resource.

ID	Nombre	Type	Capacity	Schedule Name	Schedule Rule	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	Mostrador AA	Fixed Capacity	6		Post	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
2	Maquina de Rayos X	Fixed Capacity	2		Post	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
3	Mostrador de Migraciones	Fixed Capacity	6		Post	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
4	Mostrador PU	Based on Schedule	Agenda check in PU	Agenda check in PU	Wait	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
5	Mostrador GO	Fixed Capacity	3		Post	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
6	Mostrador AR	Fixed Capacity	3		Post	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
7	Mostrador PZ	Fixed Capacity	3		Post	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
8	Mostrador BR	Fixed Capacity	2		Post	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
9	Mostrador LA	Fixed Capacity	3		Post	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
10	Mostrador BI	Fixed Capacity	6		Post	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
11	Mostrador LIA	Fixed Capacity	6		Post	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

Figura 19 Módulo de datos Resource (recurso)

En la figura 19 vemos el listado de los módulos de datos creados para nuestro modelo. Allí vemos todos los mostradores de cada una de las aerolíneas que operan en el aeropuerto. Tal cual se mencionó en la sección 4.2.2. tomamos el criterio IATA de apertura de mostradores que para aviones narrow-body es de tres mostradores por vuelo y para aviones wide-body es de seis mostradores. Así se puede observar en la figura cómo se han modelado, ya que en el campo Capacity vemos la cantidad asignada a cada aerolínea. Sólo vemos dos excepciones que son la aerolínea Sol (8R, por su código IATA), que tiene sólo 2 mostradores por utilizar aviones muy pequeños de hasta 40 plazas; y Pluna, que por tener gran canti-

dad de vuelos no se aplica el criterio del tamaño de avión y generamos un Schedule que varía según las necesidades de la hora del día entre 3 y 6 mostradores. También podemos observar en la figura 19 que no solo hemos definido mostradores de check-in como recursos en este modelo sino que también hemos generado los recursos necesarios para los procesos posteriores de seguridad y migraciones.

El último módulo de Arena utilizado para modelar el proceso de check-in, también se puede ver en la figura 18, y tiene que ver con el tiempo en que se produce el mencionado retraso del módulo Process. Normalmente, los tiempos de proceso en una simulación, así como los arribos de entidades, también son generados aleatoriamente. Es decir, debemos modelar una distribución probabilística para que los tiempos de retraso en un proceso sean generados en función de ella. Para esto Arena prevé una serie de distribuciones que son las más comunes para este tipo de aplicaciones y la pone a disposición en el menú de configuración del módulo Process a través del campo Delay Type que se puede ver en la figura 18. Además de proveer una serie de distribuciones estándar, Arena nos permite generar nuestras propias expresiones lógicas o matemáticas para definir el tiempo de retraso a generar en ese proceso. Ya sea porque las provistas no cubren nuestro caso y necesitamos usar otras distribuciones probabilísticas, o porque queremos generar nuestra propia expresión lógica o matemática para generar los tiempos de retraso. Este es nuestro caso, usaremos una expresión, y para esto recurriremos al módulo Expression del Advanced Process Panel. Allí podemos definir la expresión y darle un nombre y luego invocarla en el modelo a través del mismo como si fuese una función o una variable en un lenguaje de programación tradicional.

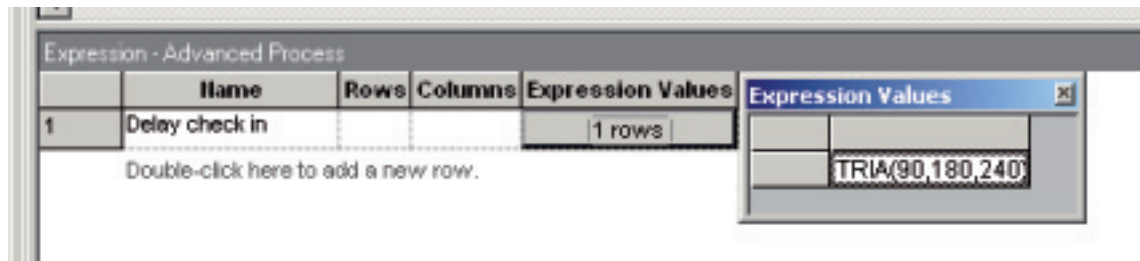


Figura 20 Módulo Expression de nuestro modelo

En nuestro caso hemos utilizado para nuestra expresión una de las distribuciones más comunes y al mismo tiempo una de las propuestas en el menú del módulo Process como tiempo de retraso de la entidad en el proceso. Esta es la distribución triangular, que como vemos en la figura 20 tiene 3 parámetros, una media, un valor mínimo y uno máximo. La media será el valor con la mayor probabilidad de ocurrencia y los valores mínimo y máximo serán los límites de las posibles ocurrencias, ya que los valores fuera de ese rango tienen probabilidad de ocurrencia cero. Es decir, la distribución sólo genera números dentro del rango y la mayor probabilidad de ocurrencia la tiene la media, y a partir de ella las probabilidades de ocurrencia disminuyen hacia los extremos del mismo. A pesar de ser esta una distribución muy utilizada para describir retrasos en procesos, hemos recurrido a una expresión en lugar de ingresarla directamente en el módulo Process del check-in porque esto da flexibilidad y facilidad en la parametrización del modelo. Hemos decidido modelar el proceso de check-in con los mismos parámetros para todas las aerolíneas, estos son los que vemos en la figura 20, un promedio de 3 minutos (180 segundos) un tiempo mínimo de un minuto y medio (90 segundos) y uno máximo de 4 minutos (240 segundos). Por eso usando el nombre del módulo Expression (Delay check in) dentro de cada módulo Process de check-in, podemos realizar cambios en la configuración una sola vez en el módulo Expression, en lugar de tener que configurar uno por uno los nueve módulos Process de check-in.

Hasta ahora hemos visto como se generan los pasajeros de nuestro modelo en los módulos Create según distribuciones probabilísticas de tipo Poisson no estacionarias, luego como entran en las colas de check-in hasta ser atendidos en un mostrador tomando para eso el uso de un recurso mostrador durante un tiempo que tiene una distribución triangular con una media de 180 segundos y un mínimo y máximo de 90 y 240 segundos respectivamente. Siguiendo con nuestro modelo, el próximo proceso a llevarse a cabo es el de chequeo de seguridad. Pero antes de eso, encontramos en nuestro modelo (figura 17) un módulo de tipo Assign, perteneciente al Basic Process Panel. Arena permite, para dar flexibilidad a la generación de modelos, el uso de variables. Estas pueden ser, tanto generadas por el usuario, como internas de Arena. Las generadas por el usuario son aquellas que se crean a través de los módulos de datos tipo Variable y podemos guardar en ella tanto valores del modelo como resultados de cálculos o expresiones. Las variables internas de Arena son aquellas que están definidas dentro del sistema y a las

que podemos acceder para obtener información sobre el modelo o la simulación. El módulo Assign, que mencionamos arriba se utiliza para dar valor a las variables generadas en el módulo de datos Variable. En la figura 21 vemos el módulo Assign de nuestro modelo. En él tenemos tres variables y sus valores para ser asignados a ellas.



Figura 21 Configuración del módulo Assign de nuestro modelo

Todas las variables que se utilizan aquí deben estar previamente definidas en el módulo de datos Variable del Basic Process Panel (que se ve en la figura 22). En nuestro modelo vamos a usar estas variables para generar datos útiles para nuestras estadísticas, y la animación del mismo. Una de las variables definidas es, Total colas, que suma los valores de cada una de las colas de check-in de cada aerolínea para obtener el número total de pasajeros haciendo colas de check-in en un determinado momento. Otra es, Total cola seguridad, que guarda el número de pasajeros que se registra haciendo la cola de seguridad en un determinado momento. Y, por último, Total cola migraciones, que hace lo mismo que las anteriores para el proceso de migraciones.

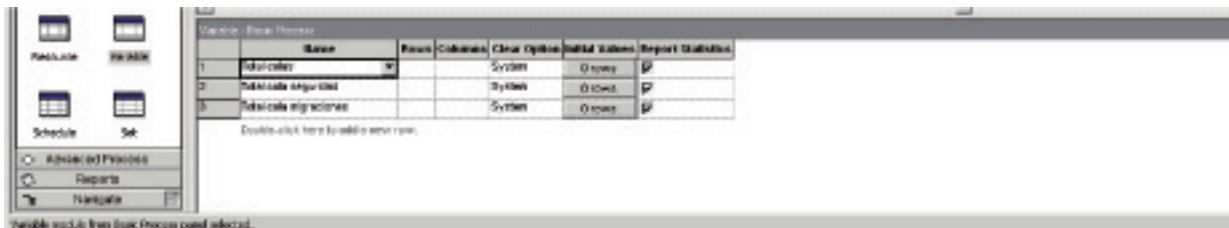


Figura 22 Módulo de datos Variable de nuestro modelo

Si observamos la figura 21, veremos que para poder guardar estos valores en las variables estamos recurriendo a una función de SIMAN (el lenguaje de simulación usado por Arena) que devuelve el número de entidades en una cola. Es la función NQ().

Siguiendo con el recorrido del diagrama de flujos que deben realizar las entidades, entramos en el módulo tipo Process del proceso de seguridad. Así como los usados para representar el proceso de check-in, este módulo permite a las entidades realizar la operación de tomar un recurso, usarlo y luego liberarlo. El recurso definido en el módulo Resource (ver figura 19) para el proceso de seguridad es la Máquina de rayos X. Hemos definido para nuestro modelo la presencia de dos máquinas de rayos con las cuales el personal de seguridad realiza el chequeo, por eso vemos que el campo Capacity de la figura 19 dice 2. Y para el tiempo de esta operación hemos definido una distribución triangular con una media de 30 segundos, un mínimo de 20 segundos y un máximo de 60 segundos. Estos tiempos han sido relevados y proporcionados por el Departamento de Infraestructura del aeropuerto de Carrasco y corroborados por recomendaciones IATA y por Takakuwa y Oyama (2003) que describen tiempos similares en su estudio.

El último proceso que atraviesa una entidad antes llegar al módulo Dispose, que se encargará de su destrucción, representando la salida de la entidad de la frontera del sistema, es el de migraciones. Este es un módulo Process del Basic Process Panel así como el de check-in y seguridad. Al igual que estos últimos, cuenta con un recurso definido en el módulo de datos Resource llamado Mostrador de Migraciones, que vimos en el listado de la figura 19. Allí vemos que se definió la capacidad del recurso en 8, ya que el diseño de la nueva terminal prevé ese número de posiciones para realizar el trámite de migraciones. También en este caso hemos configurado la acción como Seize, Delay, Release, porque cuando un pasajero llega a un mostrador este ocupa al recurso (empleado y mostrador), realiza el proceso representado por el retraso (delay) y libera el recurso. De la misma forma que en los procesos de check-in y migraciones, hemos elegido una distribución triangular para representar el tiempo del proceso de migraciones. En este caso con una media de 120 segundos, un mínimo de 60 y un máximo de 180 segundos. En la figura 23 vemos el cuadro de diálogo usado para configurar el proceso.

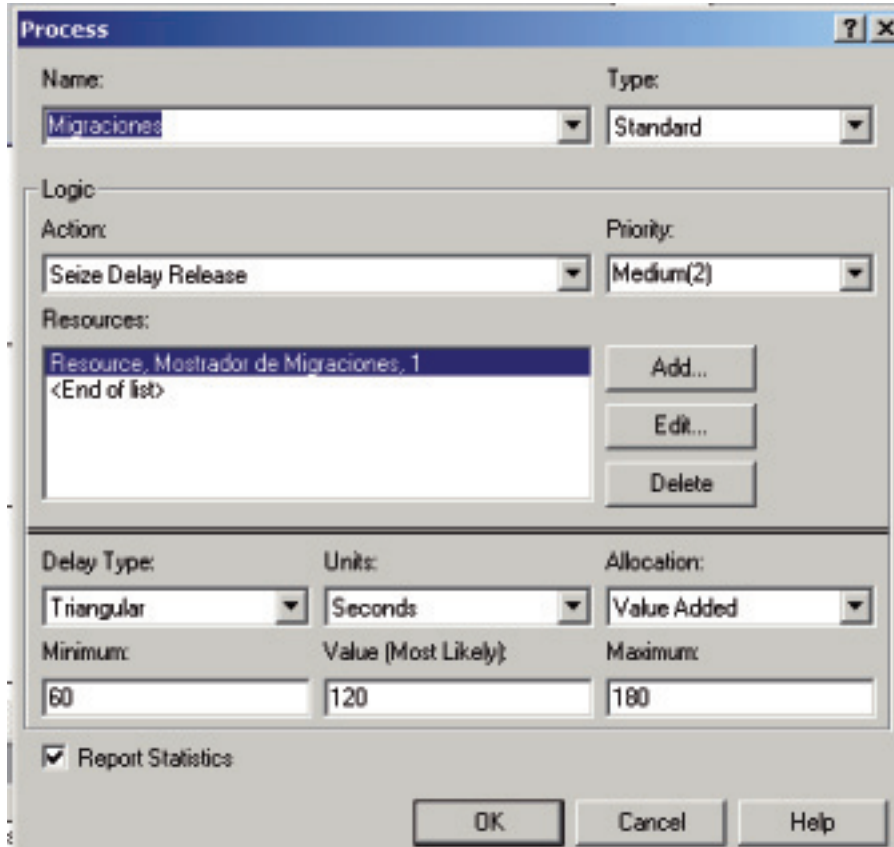


Figura 23 Configuración del módulo Process de migraciones y de seguridad

Como mencionamos anteriormente, luego de este proceso sólo queda que el pasajero (la entidad) salga del modelo y sea destruida. En el caso del pasajero, se dirigirá hacia el salón de embarque correspondiente, para abordar su avión, que en nuestro modelo esta fuera de su frontera.

Existe un aspecto más de nuestro modelo que no hemos explicado hasta ahora. Para mostrar una de las funciones avanzadas de Arena, hemos recurrido al uso de un módulo del Advanced Process Panel para generar estadísticas a la medida de nuestro proyecto y relacionadas con su objetivo que normalmente no se encuentra en los reportes estándar. El módulo Statistic nos permite generarlas. En la figura 24 vemos la configuración de dicho módulo.

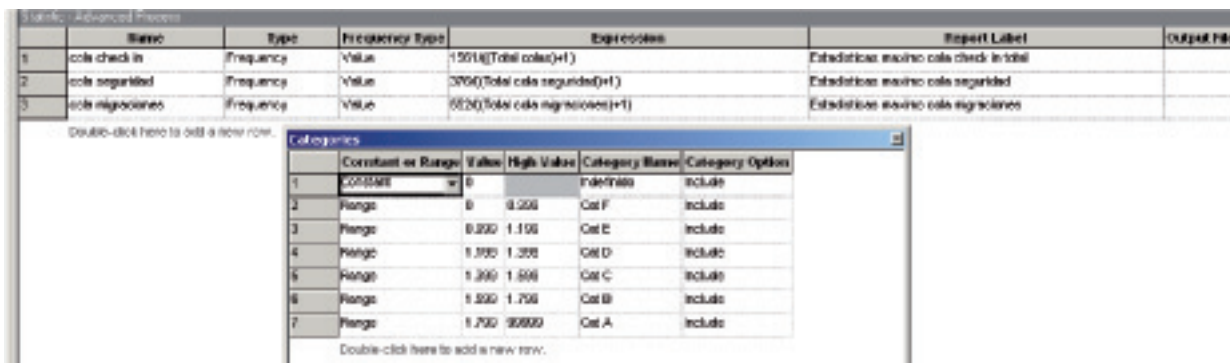


Figura 24 Módulo Statistic configurado y categorías para la cola de check-in

Allí hemos definido el tipo de estadística Frequency, esta nos permite recolectar datos sobre el des- involucramiento en el tiempo de una variable, una expresión o el estado de un recurso. En nuestro caso vamos a usar las variables que son ajustadas por el módulo Assign que vimos anteriormente. Para esto debemos definir categorías, estas son rangos de valores que nuestra variable puede tomar. Una vez terminada la simulación tendremos un reporte específico indicando que porcentaje del tiempo de la simulación, la variable estuvo en cada uno de los rangos definidos. Como dijimos antes y observamos en la figura 24, usamos las variables que contienen la cantidad de entidades en cada una de las colas

que nos interesan para el objetivo del estudio. Estas son Total colas, que contiene la suma de todas las colas de check-in, Total cola seguridad, que contiene el número de pasajeros esperando en cola para el proceso de seguridad y Total cola migraciones que hace lo mismo para el proceso de migraciones. En cada caso estamos generando una expresión que usa estas variables para calcular los ocupantes por metro cuadrado de cada una de las áreas de cola de los procesos mencionados. Luego se los compara con las categorías definidas para determinar en cuál se encuentra. Los datos se acumulan para generar el reporte una vez terminado el experimento de simulación. A modo de ejemplo vamos a analizar la expresión de la cola de check-in:

$$1561/((\text{Total colas})+1)$$

El número 1561 proviene de la tabla de la figura 11 y es la cantidad de metros cuadrados del área de colas de check-in, el cual dividimos por la cantidad de pasajeros que se encuentran en dichas colas para obtener el número de ocupantes por metro cuadrado. Hemos agregado la suma del valor 1 para que el compilador de Arena no de un mensaje de error por una división por cero, ya que las variables se inicializan automáticamente en ese valor y el valor es despreciable respecto a las estadísticas como para influir en su resultado. Asimismo, en la figura 24 vemos el listado de categorías para el reporte de estadísticas de las colas de check-in, que es básicamente una transcripción de la fila que tiene los valores de cada categoría IATA para las áreas de check-in de la figura 10 de este trabajo.

Hecho esto, estamos en condiciones de correr nuestro modelo y directamente acceder al los reportes de frecuencias de Arena para tener la respuesta objetivo de este trabajo. En qué categoría IATA de nivel de servicio se encuentran las áreas de colas de check-in, seguridad y migraciones de la nueva terminal del aeropuerto de Carrasco.

4.3.4. Animación

Como mencionamos en secciones anteriores, la animación de una simulación es una herramienta útil de comunicaciones. Tanto para quienes trabajan en el desarrollo del modelo, que la utilizan para entender de manera gráfica el mismo, como para quienes son expertos en el sistema real, que pueden usarla para ayudar a validar y verificar el modelo.

Para el modelo desarrollado en este trabajo, hemos generado una serie de elementos utilizando las herramientas que Arena proporciona para la inclusión de animación y ayudas gráficas en los modelos.

En primer lugar, hemos asignado a las entidades un gráfico que representa a una persona para hacer más clara la identificación de las mismas. En la figura 25 vemos la pantalla de Arena durante la ejecución de una simulación donde se observan las gráficas representando a las entidades que forman cola en el módulo del check-in de Pluna.

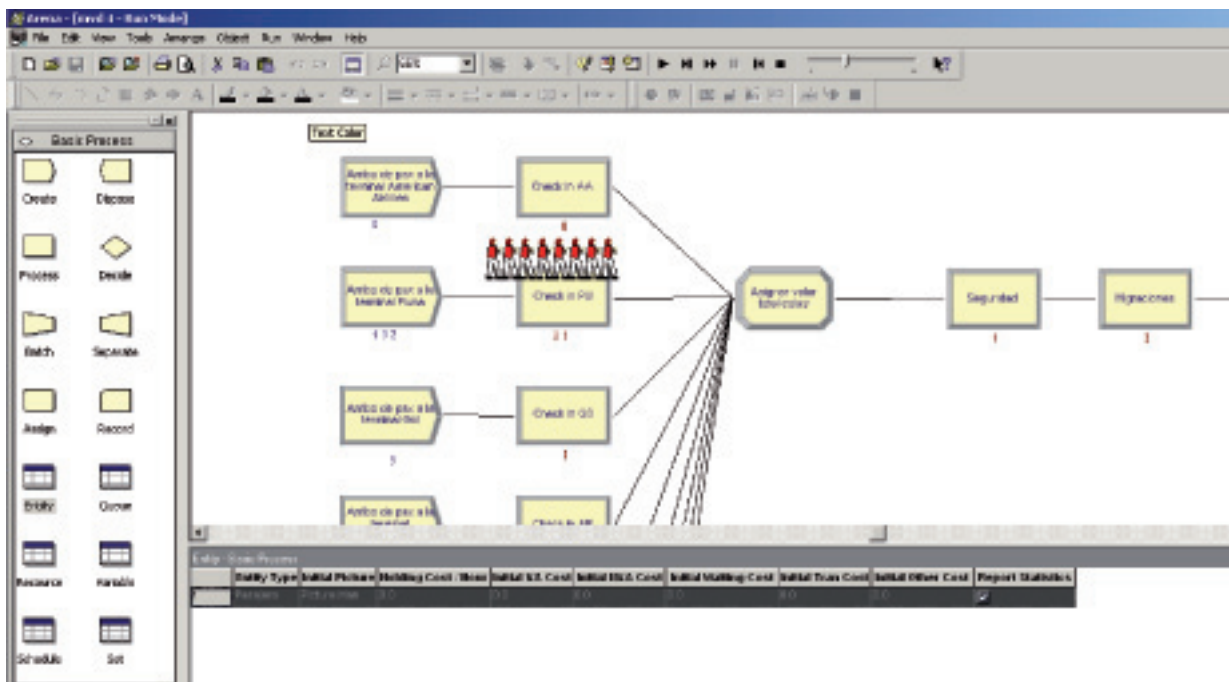


Figura 25 Configuración de la entidad Pasajero y su gráfica

También, en la parte inferior de la figura vemos la configuración de la entidad de nuestro modelo y el campo Inicial Picture (figura inicial) en el cual figura el nombre de la gráfica elegida, Picture.Man.

El otro elemento de animación que incluimos en nuestro modelo es el de las gráficas. Estas tienen en el eje x el tiempo de simulación en minutos (1440, 24 horas) y en el eje y, la cantidad de pasajeros en la cola de check-in para el de la izquierda y la cantidad de pasajeros en la cola de seguridad para la de la derecha. En la figura 26 observamos como se ven luego de ejecutada la simulación.

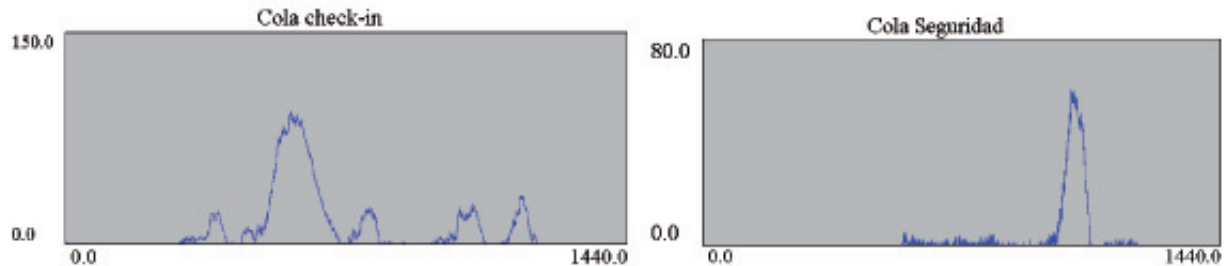


Figura 26 Gráficas de las colas del sistema

Arena permite crear este tipo de animaciones de manera muy simple. En este caso solo tenemos que generar el gráfico a través de su herramienta de animación y configurar unos pocos parámetros. En la figura 27 vemos el cuadro de dialogo usado para configurar la gráfica de la izquierda. En él encontramos los parámetros más importantes, uno de ellos es la o las expresiones a graficar, que en nuestro caso es solo una, la variable Total colas, que contiene la suma de todas las colas de check-in. Y el otro más importante es el que dice Time Range, donde definimos en qué período de la simulación queremos que se dibuje la gráfica. El resto de los parámetros tiene que ver con la estética de la gráfica.

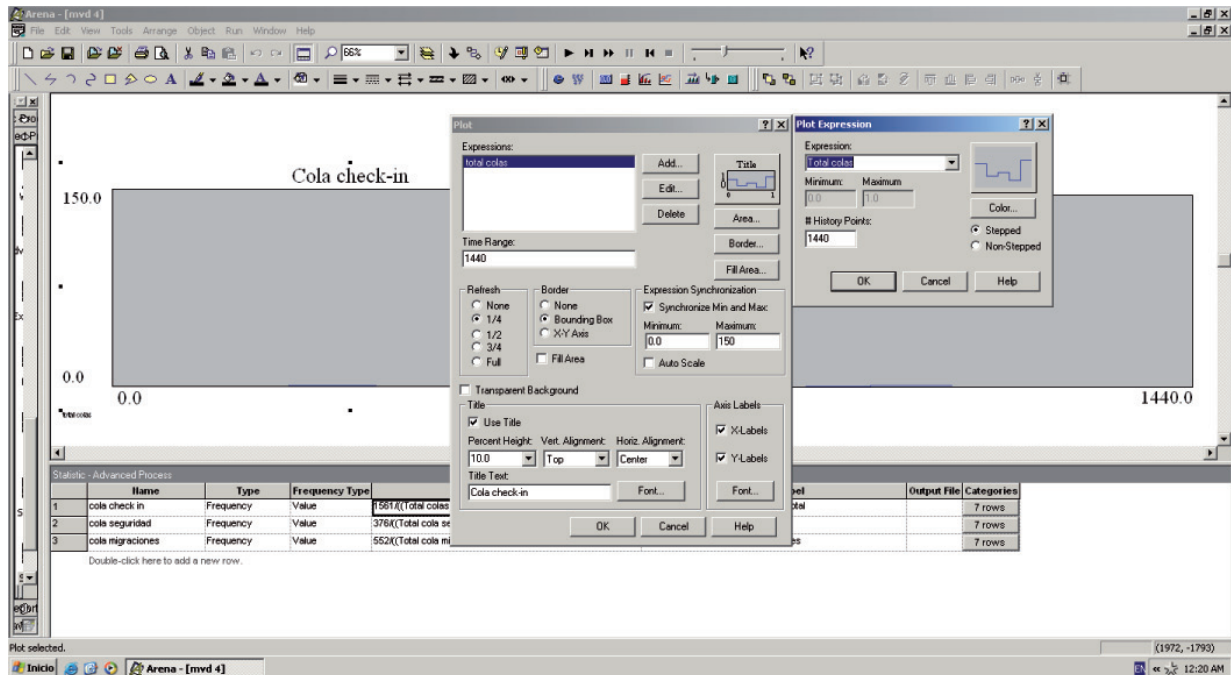


Figura 27 Configuración de gráfica de animación

Como podemos observar hemos creado herramientas visuales para poder seguir y comprender el modelo y la evolución de la simulación de manera muy simple y con muy poco esfuerzo. Sin embargo, Arena provee métodos y herramientas para realizar animaciones muy complejas y potentes.

4.4. Simulación

En la sección 3.3. de este trabajo presentamos las opciones más importantes que brinda Arena para la configuración de la ejecución de un experimento de simulación una vez que el modelo está terminado.

Para nuestro modelo hemos configurado algunos aspectos necesarios para su correcta ejecución. El

primero, y posiblemente el más importante, es el llamado Replication Parameters. En la figura 28 vemos la ventana por donde accedemos a éste.

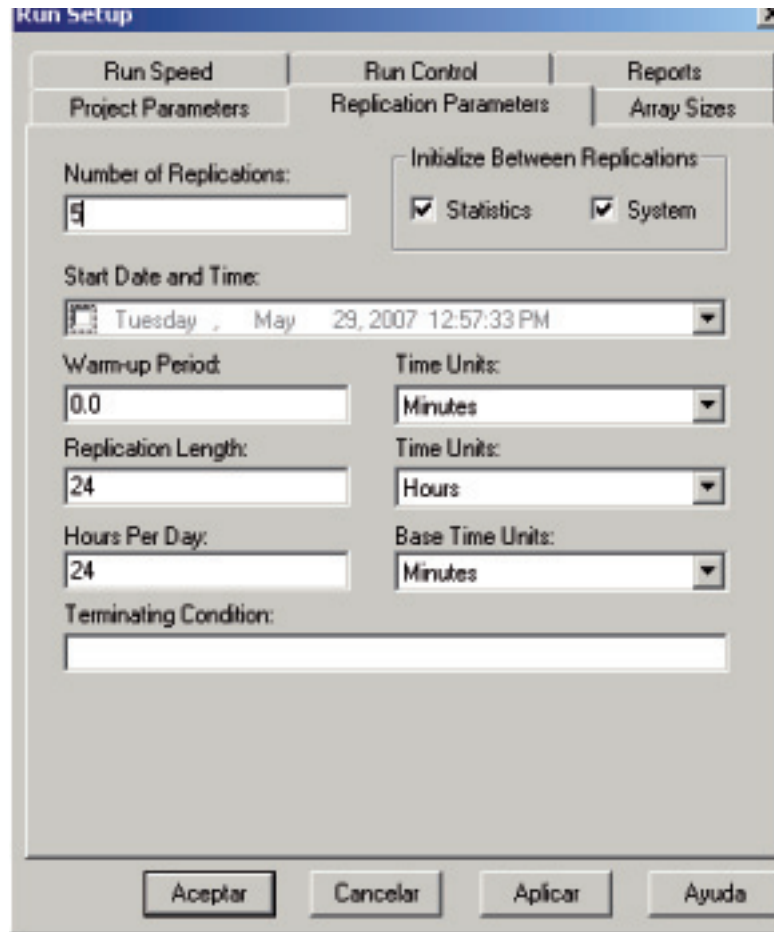


Figura 28 Replication Parameters para nuestro modelo

Este panel de configuración nos permite controlar el tiempo de ejecución de la simulación así como especificar una condición de fin del experimento y la cantidad de réplicas a correr. Para nuestro proyecto hemos definido que simularemos un día completo, por eso vemos en la figura que el Replication Length está fijado en 24 horas. Otro parámetro importante es el Base Time Units, en el que fijamos la unidad de tiempo que Arena usará en todos los reportes de la simulación. Como último parámetro, hemos elegido la cantidad de réplicas a ejecutar de la simulación de las cuales se recolectarán datos estadísticos para presentar en los reportes del experimento. En este caso hemos elegido realizar 5 réplicas del experimento, seguramente sería mejor usar un número mayor, para poder obtener intervalos de confianza más pequeños en los reportes, pero por usar la versión limitada de Arena no es posible.

Siguiendo con la configuración de nuestra simulación, vamos a modificar algunos parámetros más en la pestaña Project Parameters del mismo menú, el Run Setup. En primer lugar, debemos nombrar al proyecto, ya que el nombre asignado en este lugar será el usado en todos los reportes y documentos que emita Arena sobre este modelo. Esto se define en el campo Project Title, y para nuestro modelo hemos elegido el nombre "Nueva Terminal de Carrasco". A continuación se debe ingresar el nombre del responsable del proyecto y una descripción del mismo. También encontramos en esta ventana de configuración un elemento importante del experimento de simulación, la definición de qué estadísticas debe generar Arena de nuestro modelo. Para nuestro modelo hemos seleccionado todos aquellos elementos que nos interesan y que participan del modelo. Estos son: las entidades, los recursos, las colas y los procesos. En la figura 29 vemos todos estos parámetros y sus respectivos campos.

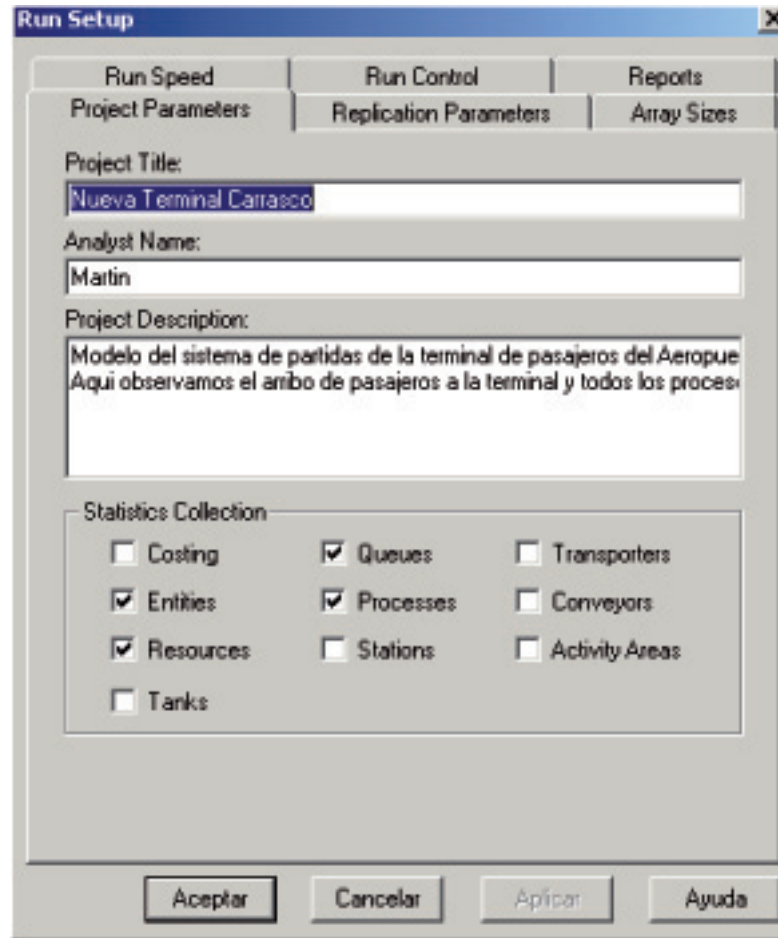


Figura 29 Configuración de los Project Parameters de nuestro modelo

Cabe mencionar un último aspecto de configuración del modelo necesario para la ejecución. Como hemos dicho en la sección 4.3.2. tenemos una limitación respecto a las entidades que se generen y convivan en el modelo de manera concurrente. Debido a que solo pueden ser 150, hemos tenido que hacer algunos ajustes al modelo original para que Arena nos permita la ejecución completa del experimento de simulación, es decir, de principio a fin. Por esto, y luego de varios intentos de ejecución fallidos, llegamos a la conclusión que debíamos “apagar” la generación de entidades (pasajeros) de las aerolíneas Gol, TAM y LAN.

En la figura 30 vemos el listado de los módulos Create de nuestro modelo donde se observa que para dichas compañías aéreas hemos asignado una distribución exponencial con una media de una hora a la generación de entidades. Esto tendrá un efecto de disminución sobre la entrada de pasajeros al sistema, a uno por hora, sin la necesidad de modificar el modelo quitando los módulos Create mencionados.

Luego de realizados estos ajustes, estamos en condiciones de ejecutar la simulación y llevar a cabo el experimento.

4.5. Resultados

A partir del modelo descrito en la sección 4.3. y la configuración del experimento que se muestra en la sección anterior, hemos ejecutado la simulación. De los reportes generados por Arena hemos seleccionado los más significativos para ser volcados en la siguiente tabla:

Resultado	Valor	Intervalo de confianza 95%
Cantidad total de pasajeros procesados (promedio de 5 réplicas).	1739	29,15
Tiempo promedio de paso por el sistema. Por pasajero.	13,22 min.	1,72
Tiempo máximo de paso por el sistema. Por pasajero.	48,70 min.	
Tiempo promedio de espera por pasajero.	7,78 min.	1,71
Tiempo máximo de espera por pasajero.	42,68 min.	
Tiempo promedio en proceso por pasajero.	5,44 min.	0,01
Tiempo máximo en proceso por pasajero.	7,43 min.	
Cantidad máxima de pasajeros en colas de check-in	90	
Cantidad máxima de pasajeros en colas de seguridad	86	
Cantidad máxima de pasajeros en colas de migraciones	2	

En el Anexo C de este trabajo encontramos el reporte completo de Arena del cual fueron extraídos los datos presentados.

En una primer mirada a estos datos, podemos llevar a cabo un proceso de validación del modelo generado. Por ejemplo, el dato más evidente es el número de pasajeros generados y procesados por el sistema, que es de 1739. Mientras que si tomamos los datos de la planilla de la figura 14, donde figuraban los vuelos del día y sus pasajeros esperados y los sumamos dejando de lado las compañías Gol, TAM y LAN, llegamos a un resultado de 1735. Esto nos da una primera pauta de la validez del sistema, mientras que podemos tomar otras de la animación. Si observamos la figura 26 donde se grafican la cantidad de personas en las colas de check-in y seguridad vemos picos en las mismas que podemos comparar con observaciones del sistema real. En las cola de check-in vemos un pico muy claro alrededor de las 10 AM, que también observamos en el sistema real. En cuanto a la de seguridad, ahí podemos observar otro pico alrededor de las 6 PM que también es observado en el sistema real.

Siguiendo con el análisis de los resultados de la simulación, vamos a detenernos en el tiempo máximo de paso por el sistema, que pareciera ser alto incluso para un valor máximo. Evidentemente la influencia más grande de ese valor, de 48,7 minutos es el tiempo de espera máximo que registra un valor de 42,68 minutos. Para entender porque se genera ese valor, vamos a investigar los tiempos máximos de espera en cada una de las colas del sistema. Allí encontramos que en la cola de Pluna tiene valor de 42,46 minutos que evidentemente influencia de manera definitiva el tiempo máximo de paso por el sistema de nuestro experimento. Este dato es importante, sin embargo un problema de colas de check-in de una aerolínea no es algo preocupante, ya que los mostradores pueden ser asignados dinámicamente y que con solo agregar más se corregiría ese tiempo prolongado.

Otro valor observado en el reporte que resulta interesante es el del número de pasajeros y tiempo de espera de los mismos en el proceso de seguridad. Normalmente se toma como cuello de botella y lugar conflictivo en el sistema de partidas al proceso de migraciones, pero como se ve claramente en los datos presentados la cola de migraciones tiene un máximo de dos pasajeros registrados, mientras que la de seguridad presenta ochenta y seis. También observamos que el tiempo de espera máximo en el proceso de seguridad es de 26,86 minutos mientras que para el proceso de migraciones es de menos de un minuto. Todos estos datos revelan que, al contrario de lo que las observaciones empíricas mostrarían, el cuello de botella en los procesos del sistema de partidas está en el de seguridad. Y que como en el sistema real nunca funcionan todos los mostradores de migraciones a la vez por falta de personal, la percepción es que este es el verdadero cuello de botella del sistema cuando no lo es.

Un aspecto a tener en cuenta a partir de los resultados de la simulación es de cómo se comparan los datos obtenidos con estándares y recomendaciones. Para los tiempos de proceso de cada una de las partes del sistema aeroportuario, IATA tiene recomendaciones sobre los máximos aceptables. En la siguiente tabla volcamos algunos tiempos obtenidos para su comparación con las recomendaciones:

Tiempo máximo cola Pluna	42,46 min.
Tiempo máximo cola Aerolíneas Argentinas	37,41 min.
Tiempo máximo cola seguridad	26,62 min.

La elección de los valores de la tabla se hizo porque son aquellos que están fuera de rango. Es decir, IATA recomienda que el tiempo máximo de cola de check-in no supere los 30 minutos, mientras Aerolíneas Argentinas y Pluna han acumulado picos en que lo hacen. Para el proceso de seguridad IATA prevé un tiempo máximo de 7 minutos mientras que el máximo del modelo es mayor. Sin duda existen explicaciones a estos fenómenos que podrían encontrarse estudiando el modelo con mayor profundidad y revisando las especificaciones IATA, ya que el manual consultado es el último publicado, en 2004.

Finalmente, vamos a analizar los datos que motivaron este estudio, los provenientes de los máximos niveles de pasajeros en las colas de check-in, migraciones y seguridad. En la primer tabla presentada en esta sección encontramos los valores de estos datos que ha arrojado el experimento de simulación. Sin realizar complicadas cuentas podemos observar a simple vista que el aeropuerto es categoría A de nivel de servicio según las normas IATA. Recordemos que el espacio designado a colas de check-in es de 1561 metros cuadrados y el máximo nivel de cola para esta área es de 90 pasajeros según la simulación. Para ser categoría A debemos tener un mínimo de 1,8 metros cuadrados por ocupante y en nuestro caso hay 17,34 metros cuadrados por ocupante. Lo mismo ocurre en las áreas de seguridad y migraciones donde se llega holgado al nivel requerido por la categoría A. En la siguiente tabla encontramos los cálculos específicos:

Área de proceso	Superficie	Máximo en cola	M ² por ocupante	M ² por ocupante mínimo IATA
Check-in	1561 m ²	90	17,34	1,8
Seguridad	376 m ²	86	4,37	1,4
Migraciones	552 m ²	2	276	1,4

Como dijimos anteriormente, esta claro que el aeropuerto es categoría A en todas las áreas estudiadas. Sumado a esto podemos ver el reporte de Frecuencias generado por Arena a partir del módulo Statistic descrito en la sección 4.3.3. que se encuentra en el Anexo C de este trabajo, donde veremos que cada una de las áreas estudiadas permanece en categoría A durante el 100% del tiempo de la simulación.

A modo de última reflexión respecto a los resultados, cabe mencionar que el presente trabajo se ha realizado de la manera más precisa posible respecto al sistema real, se han hecho suposiciones y simplificaciones, y se han usado muchos estándares de mercado en el modelado del mismo. Esto hace que recomendamos que los resultados sean tomados como los de un prototipo para el análisis, y no un modelo acabado para utilizarlo en el diseño del sistema real.

5. Conclusiones

En la sección 1.1. del trabajo se presentaron una serie de objetivos a cumplir. A lo largo del mismo hemos avanzado en los conocimientos y experiencias necesarias para poder llegar a conclusiones satisfactorias respecto a ellos.

En primer lugar se ha planteado que la simulación es de probada efectividad en el análisis de capacidades de aeropuertos. Durante todo el trabajo hemos presentado teoría y practica a través de ejemplos de sistemas y problemas en los que la simulación resulta una vía muy efectiva para llegar a respuestas y soluciones. Muchos de los ejemplos y casos prácticos explicados, a pesar de nos ser directamente relacionados a aeropuertos, podíamos de una forma u otra adaptarlos a situaciones objeto de nuestro estudio. Sin embargo, y de manera contundente, la documentación consultada para el trabajo ha revelado que la simulación esta no solamente probada como una herramienta útil a la hora de estudiar aeropuertos y sus capacidades, sino que ha sido declarada como método obligatorio para el estudio del impacto del cambio de ciertos procesos aeroportuarios (Pendergraft et al, 2004). En la sección 4.1. encontramos una serie de ejemplos concretos de bibliografía sobre casos reales de estudios de simulación realizados en aeropuertos. Entre ellas encontramos a Doshi y Moriyama (2002), Joustra y Van Dijk (2001), Verbraeck

y Valentin (2002) y Kiran et al (2000). Toda esta documentación considerada de relevancia, y otra que ha sido consultada pero de menor importancia para nuestro trabajo nos da la pauta y la certeza de que la simulación es usada asiduamente en el campo de los aeropuertos y que resulta de gran efectividad para los usuarios de la misma.

En segundo lugar, se propuso que Arena podía ser aplicado a la simulación de aeropuertos. Respecto a esto, hemos dado pasos importantes. Se ha revisado valiosa bibliografía sobre la materia. Como Takakuwa y Oyama (2003) que describen la utilización de Arena para la simulación del aeropuerto Kansai en Osaka, Japon; Gatersleben y van der Weij (1999) que describe el uso de Arena para encontrar cuellos de botella en los procesos del Aeropuerto de Schiphol en Ámsterdam, Holanda a través de la simulación; y por último destacamos el paper de Snowdon et al (1998) donde se presenta una solución genérica basada en Arena para simulación de procesos en aeropuertos desarrollada por IBM y llamada Journey Management. Además de la documentación relevada hemos construido un modelo del Aeropuerto de Carrasco que ha mostrado ser válido y útil para su estudio.

Por último, hemos usado el modelo construido en Arena para realizar experimentos de simulación y llegar a conclusiones sobre la categoría IATA de nivel de servicio en se encuentra el aeropuerto modelado en las áreas de check-in, seguridad y migraciones. Podemos afirmar que llegamos a conclusiones satisfactorias, no sólo sobre el objetivo de determinar la categoría del aeropuerto sino que hemos encontrado muchos puntos de análisis útiles en los reportes generados por el experimento.

A partir de todo lo expuesto, podemos sostener que hemos alcanzado y probado los objetivos planteados al inicio del trabajo y que, más allá de poder continuar la investigación y refinación del modelo, tal como veremos en la próxima sección, nos encontramos satisfechos con el resultado de esta tesina.

6. Líneas de investigación futuras

Para poder continuar esta investigación hacia un desarrollo mayor, el primer paso que se debería dar es el de cambiar la versión de Arena. Habría que pasar de la académica a la comercial para poder usarla en todo su potencial y dejar de tener la limitación en cantidad de entidades concurrentes. Así podríamos modelar casos de estrés en los que sobrecarguemos al aeropuerto para analizar su comportamiento.

Otro caso interesante para continuar el desarrollo del modelo presentado en el trabajo puede ser el de generar un modelo de optimización de recursos para encontrar una combinación horaria de asignación de los mismos en cada proceso de manera de evitar los picos de colas y mantener determinada categoría de nivel de servicio. Esto podría hacerse usando la herramienta OptQuest de Arena, u otros métodos de búsqueda estocástica como algoritmos genéticos o sistemas evolutivos.

Sería también una evolución lógica a nuestro modelo la de generar un sistema por el cual dando como entrada una programación de vuelos y pasajeros se pueda armar automáticamente la distribución Poisson no estacionaria necesaria para mostrar el arribo de pasajeros. Al mismo tiempo se podría generar una representación de la disponibilidad de recursos de check-in más realista basada en observaciones o en las políticas internas de cada una de las aerolíneas.

En fin, existen innumerables casos en los cuales este trabajo puede resultar útil y por lo tanto es posible expandirlo. Claramente podemos mirar los límites impuestos por nosotros mismos cuando definimos el trabajo como punto de partida para ampliar diversos aspectos del modelo. Asimismo, esperamos que este trabajo sea de utilidad para el lector así como lo fue para nosotros y que sea fuente de inspiración para futuros desarrollos.

7. Glosario

Wide body: Avión de gran porte. Se considera el Boeing 767 el más pequeño de esta categoría. También se consideran en esta categoría el Boeing 777, 747 y el Mac Donnell Douglas MD-11.

Narrow body: Es la categoría de aviones más pequeños, considerando al Boeing 757 el más grande de la categoría seguido por el Airbus A320.

Espigón: Así llamamos a las alas de un edificio aeroportuario que conectan a los pasajeros con los aviones

IID: De la sigla en inglés para Independent and Identically Distributed. En estadística son muestras independientes y distribuidas idénticamente.

IATA: Siglas de International Air Transport Association. Asociación que reúne a los actores de la industria aerocomercial.

OACI: Organización Aeronáutica Civil Internacional. Organización que nuclea los entes responsables de la aviación en el mundo.

ACI: Airports Council International: Organización que reúne a los operadores aeroportuarios del mundo.

ATC: Siglas en inglés de Air Traffic Control. O control de tráfico aéreo.

8. Anexo A: Módulos principales de Arena

8.1. El panel de Procesos Básicos

Veremos los módulos de flujo y datos que se usan para modelar un proceso, junto con ejemplos de uso.

8.1.1. Los módulos de flujo

Los módulos de flujo son un conjunto de objetos que, puestos en la ventana de diagrama, describen un proceso.

El módulos Crear

Este módulos pretende officar como punto de entrada de las entidades, en un modelo simulado. Las entidades se crean en este módulos basadas en un tiempo o rango de tiempos determinados. Luego dejan este módulos para empezar a viajar por el sistema. El tipo de entidad que se crea se define en este módulos.

Entre los usos mas comunes de este módulos, podemos encontrar el comienzo del proceso de fabricación de una parte, la llegada de un documento (orden, cheque, formulario) a la entrada de un proceso de negocios, la llegada de un cliente al mostrador de la recepción (restaurant, hospital, empresa).

Campos



Nombre (Name): un identificador único, que se muestra dentro del simbolo del módulos.

Tipo de Entidad (Entity Type): nombre del tipo de entidad que se generara.

Tipo (Type): forma de arribo de las entidades. Estos tipos pueden ser:

Azar (Random), que utiliza una distribución exponencial en la que el usuario especifica la media.

agenda (schedule), utiliza una distribución exponencial en la que la media se determina por el módulos.

Constante (se utiliza un numero constante)

expresión, con una lista de expresiones de varias distribuciones.

Valor (Value): determina la media en la distribución exponencial (cuando se utiliza Random), el valor constante cuando se utiliza el tipo constante.

Nombre de la agenda (schedule name): identifica el nombre de la agenda que será usada y define el patron de arribos de entidades.

expresión (Expression): distribución o valor que especifica el tiempo entre arribos, solo valido si el tipo es expresión.

Unidades (Units): unidad de tiempo usada para los tiempos entre arribos. No aplica si el tipo es Schedule.

Entidad por arribo (Entities per Arrival): numero de entidades que ingresaran al sistema con cada arribo.

Arribos Máximos (Max Arrivals): numero máximo de entidades que este módulos generara. Cuando se alcanza este numero, no se crearan mas entidades.

Primera creación (First Creation): hora de inicio para la primera entidad, no se utiliza si el tipo es Schedule.



Módulos de terminación Dispose

Este módulos pretende servir como destino final de las entidades dentro de la simulación. Entre sus usos mas comunes puede estar: partes que salen de una línea de fabricación, el final de un proceso de negocios, clientes que salen de un negocio.

Campos

Recolectar estadísticas (Record Entity Statistics): determina si se graban o no estadísticas respecto a las entidades, entre ellas: tiempos de espera, tiempos de transferencia, totales, costos, costos de espera, costos totales.



El módulos de Proceso Process

Este módulos es donde se hacen los procesos en la simulación. Trae opciones para el comportamiento

de los recursos y para utilizar “submodelos” o especificar algún tipo de lógica necesaria para el proceso. Los costos inherentes serán añadidos a las categorías que correspondan.

Entre sus usos típicos podemos pensar en, revisar un documento, rellenar ordenes, atender a un cliente, agujerear una parte de un objeto.

Campos

Nombre (Name): un identificador único, que se muestra dentro del simbolo del módulos.

Tipo (Type): metodo de especificar lógica dentro del módulos. “Proceso Standard” (Standard Process) significa que toda la lógica estará dentro del módulos y definida por una acción en particular. “Submodelo” (Submodel) indicara que la lógica estará jerárquicamente definida en un submodelo que a su vez puede incluir cualquier numero de módulos lógicos.

acción (Action): tipo de procesamiento que ocurriera dentro del módulos. “Delay” (retraso) indicara que el proceso se retrasara sin que por ello el recurso se vea ocupado. “Seize Delay” (Retener y retrasar) indicara que se utilizara un recurso y que habrá un retraso, pero que el recurso se liberara un tiempo mas tarde. “Seize Delay Release” (Retener, retrasar y liberar) indicara que un recurso se utilizara, seguido de un retraso y finalmente se liberara. “Delay Release” indica que un recurso que ya estaba siendo utilizado y que habrá un retraso luego del cual el recurso se liberara. Todo esto aplica solo cuando el TIPO es Standard.

Prioridad (Priority): prioridad de la entidad en este módulos a la espera de que se libere un recurso junto con otras entidades esperando por el mismo recurso. No se visualiza cuando la ACCION es “Delay” o cuando el TIPO es “Submodel”.

Recursos (Resources): lista los recursos o conjunto de recursos usados para el procesamiento. No se usa cuando la acción es “Delay” o cuando el TIPO es “Submodel”.

Tipo de retraso (Delay Type): es la distribución o metodo para especificar los parámetros de retraso. Constante y expresión requieren valores únicos, mientras que Normal, Uniforme y Triangular requieren varios parámetros.

Unidades (Units): unidad de tiempo para los parámetros de retraso.

Asignación (Allocation): determina como el tiempo y costos de proceso se asignaran a la entidad. Los procesos pueden ser considerados de valor agregado, de NO valor agregado, de transferencias, espera y “otro” y los costos asociados irán a la categoría correspondiente.

mínimo (Minimum): parametro para especificar el valor mínimo para las distribuciones uniforme y triangular.

Valor (Value): especifica la media para las distribuciones normales, el valor para un retraso constante, o el modo para una distribución triangular.

máximo (Maximum): el valor máximo para las distribuciones uniforme y triangular.

Desviación Standard (Std Dev): desviacion estandard de la distribución normal.

expresión (Expression): especifica una expresión cuyo valor se evalua y se usa para el retraso de proceso.

Reporta estadísticas (Report Statistics): especifica si se recolectan o no estadísticas automaticamente y se guardan en la base de reportes de este proceso.

El campo **Recursos** (Resources) tiene su propio cuadro de dialogo:

Tipo (Type): especificacion de un recurso en particular, o seleccion de un conjunto de recursos.

Nombre del Recurso (Resource Name): nombre del recurso que será utilizado/liberado. Solo aplica si el TIPO es “resource”

Nombre del Conjunto (Set Name): nombre del conjunto de recursos del cual un miembro será utilizado/liberado. Solo aplica si el TIPO es “Set” (conjunto)

Cantidad (Quantity): numero de recursos de un tipo o de un conjunto que serán utilizados/liberados. Para conjuntos, este valor especifica el numero de recursos dentro del conjunto que serán utilizados/liberados, no la capacidad del conjunto en si.

Regla de Selección (Selection Rule): metodo de seleccionar entre dos recursos disponibles en un conjunto. “Cyclical” (cíclico) ira pasando por todos los miembros disponibles. “Random” (azar) elegirá uno al azar. “Preferred Order” (orden de preferencia) seleccionara siempre el primer miembro disponible.

Guardar (Save Attribute): atributo que se usa para guardar el índice del miembro dentro del conjunto.

Índice (Set Index): el numero de índice del miembro requerido. Cuando la acción esta definida como “Delay Release”, el valor especificado en la acción será el numero de índice del miembro que será liberado.



Módulos de Decisión Decide

Este módulos permite la toma de decisiones en el sistema. Incluye opciones para tomar decisiones

sobre una o mas condiciones o en una o mas probabilidades. Las condiciones pueden estar basadas en atributos de una entidad (como la prioridad), variables (numero de rechazos), el tipo de entidad o usando alguna expresión.

Cuando el tipo del módulos esta definido como Condición doble (2-way chance or 2-way condition), hay dos únicos puntos de salida: verdadero y falso. Cuando el tipo esta definido como múltiples condiciones (N-way chance or condition), habrá un punto de salida para cada condición y una salida que represente el "otros" (aquellas condiciones no representadas en las demás opciones).

Los usos mas típicos de este módulos son por ejemplo, el despacho de una parte defectuosa para que se fabrique de nuevo, separar cheques aceptados y rechazados, enviar clientes prioritarios a un proceso dedicado.

Campos

Nombre (Name): un identificador único, que se muestra dentro del simbolo del módulos.

Tipo (Type): Cuando el tipo del módulos esta definido como Condición doble indica que la decisión esta basada en una condición ($X > Y$) o en un porcentaje (60% si, 40% no). Cuando el tipo esta definido como múltiples condiciones (N-way chance or condition), habrá un punto de salida para cada condición y una salida que represente el "otros" (aquellas condiciones no representadas en las demás opciones).

Condiciones (Conditions): define una o mas condiciones usadas para direccionar las entidades hacia diferentes módulos. Solo aplica cuando el tipo es N-way

Porcentajes (Percentages): define uno o mas porcentajes usadas para direccionar las entidades hacia diferentes módulos. Solo aplica cuando el tipo es N-way

Porcentaje Verdadero (Percentage True): determina el porcentaje de entidades que saldrán por el punto de salida verdadero.

Si (if): tipos de condiciones disponibles para evaluacion: variable, array de variables (bidimensional), atributos, tipo de entidad, expresión.

Nombrado (Named): especifica el nombre de una variable, atributo o tipo que será evaluado cuando una entidad entra al módulos. No se aplica si el tipo es Expresión.

Es (Is): evaluador de la condición. Aplica solo a condiciones de atributo o variable

Fila (Row): especifica el índice de la fila de un array de variables. Aplica solo cuando el Tipo es N-Way o 2-way y si la variable es un array unidimensional o bidimensional.

Columna (Column): especifica el índice de la columna de un array de variables. Aplica solo cuando el Tipo es N-Way o 2-way y si la variable es un array unidimensional o bidimensional.

Valor (Value): Expresión que será comparada contra la variable o el atributo o que será evaluada para determinar la veracidad o falsedad de una condición. No se aplica a la condición de Tipo de Entidad. Si el tipo es Expresión, este valor debe incluir un evaluador (color<>Rojo)



Módulos Batch Batch

Este módulos es un mecanismo de agrupación dentro del modelo. Los batch pueden ser agrupados temporalmente o permanentemente. Los batch temporales deben ser separados usando un módulos separador.

Los batch se crean con cualquier cantidad de entidades entrantes, agrupadas según un atributo y se acumulan hasta que la cantidad deseada se alcanza. Cuando eso sucede una nueva entidad que representa la agrupación se crea.

Entre los usos mas comunes se encuentran el de juntar varias partes antes de comenzar un proceso de armado, reunir varias copias de un documento, juntar un paciente con su historia clínica antes de comenzar una cita con el doctor.

Campos

Nombre (Name): un identificador único, que se muestra dentro del simbolo del módulos.

Tipo (Type): forma de juntar entidades.

Tamaño del grupo (Batch Size): cantidad de entidades que se agruparan

Criterio de guardado (Save Criterion): metodo para asignar atributos a la entidad representativa del batch.

Regla (Rule): determina como se agruparan las entidades entrantes. "Any Entity" (cualquier entidad) ira agrupando entidades hasta alcanzar el numero determinado por el parámetro Size. "Attribute" usara un atributo especifico para agrupar las entidades. Por ejemplo si el atributo es el color, todas las entidades deben tener el mismo color para agruparse, si así no fuera, quedaran estacionadas en el módulos hasta

conseguir la cantidad necesaria para formar un batch.

Nombre del atributo (Attribute Name): nombre del atributo que se usara para agrupar. Solo se usa si la regla es "attribute".



Módulos Separador Separate

Este módulos se usa para clonar una entidad en varias o para separar un batch creado en algún punto anterior del modelo. Existen reglas para asignar costos y tiempos a los duplicados. Cuando se separa un batch, la entidad representativa que los agrupaba se elimina y se recuperan todas las entidades que la formaban, prosiguiendo en el mismo orden en el que ingresaron al batch. Cuando se hace duplicación, se crean las copias y se envían desde este mismo módulos, junto con la entidad original.

Se usa comúnmente para enviar documentos a, por ejemplo, deposito y facturación, para que se prepare la orden y se la facture al mismo tiempo, para separar un conjunto lógico de documentos que fue procesado antes.

Campos

Nombre (Name): un identificador único, que se muestra dentro del simbolo del módulos.

Tipo (Type): metodo para separar la entidad entrante. "Duplicate Original" (duplicar original) simplemente toma la entidad y la duplica una cantidad de veces definida. "Split existing Batch" (Separar un batch) requiere que la entidad entrante sea un batch creado previamente por un módulos batch.

Porcentaje de Costo de los Duplicados (Percent Cost to Duplicates): asignación de costos y tiempos de la entidad entrante a los duplicados salientes. Esta valor esta dado por un porcentaje del costo y tiempo de la entidad original. El porcentaje indicado será repartido en partes iguales entre los duplicados y el porcentaje restante lo retendrá la entidad original. Este parámetro solo es visible cuando el Tipo es "Duplicate original".

Numero de Duplicados (# of Duplicates): la cantidad de entidades clonadas que saldrán del módulos, ademas de la original. Solo se utiliza si el tipo es "Duplicate original"

Atributos de los miembros (Member attributes): metodo para determinar como se asignaran los atributos de la entidad representativa a las entidades originales. Estas opciones están relacionadas con los 6 atributos de propósito especial, a saber: type, Picture, sequence, station, jobstep y holdCostRate.

Nombre del atributo (Attribute Name): nombre del atributo de la entidad representativa que se asigna a las entidades originales del grupo.



Módulos de Asignación Assign

Se utiliza para cambiar los valores de las variables, atributos, tipos, imágenes o alguna otra variable del sistema. Se pueden hacer múltiples cambios en un solo módulos de asignación. Sirve para sumarizar las partes de semielaborados que se suman a una parte, cambiar el tipo de una entidad para representar la copia para el cliente de un formulario multipágina o cambiar la prioridad de un cliente.

Campos

Nombre (Name): un identificador único, que se muestra dentro del simbolo del módulos.

Tipo (Type): tipo de asignación.

Nombre de la variable (Variable Name): nombre de la variable a la que se le asignara un nuevo valor cuando una entidad entre al módulos. Aplica solo si el tipo es variable o array de variables.

Fila (Row) y **Columna** (Column): índice de la posición dentro del array.

Nombre del Atributo (Attribute Name): nombre de la entidad a la que se le cambiara el valor cuando entre al módulos. Solo se aplica si el tipo es Attribute.

Tipo de Entidad (Entity Type): nuevo tipo que será asignado cuando la entidad entre al módulos. Solo aplica si el Tipo es Entity Type.

Imagen de la Entidad (Entity Picture): nueva imagen que tendra la entidad cuando entre al módulos. Solo se aplica cuando tipo es Entity Picture.

Otros (Other): identifica variables especiales del sistema a las que se les cambiara el valor cuando una entidad entra al módulos.

Nuevo Valor (New Value): nuevo valor que será asignado al atributo, variable o variable especial del sistema.



Módulos de grabación Record

En este módulos se recolectan estadísticas del modelo simulado. están disponibles varias observaciones estadísticas tales como tiempo entre salidas del módulos, estadísticas de las entidades como

tiempo y costo, observaciones generales y estadísticas de intervalos. Contadores también se definen en este módulos para por ejemplo contabilizar la cantidad de trabajos terminados por hora, cuantas ordenes fueron cumplidas con retraso, controlar el tiempo que los clientes con prioridad pasan en la cola.

Campos

Nombre (Name): un identificador único, que se muestra dentro del simbolo del módulos.

Tipo (Type): tipo de estadísticas a ser generadas. "Count" incrementara o decrementará el valor de la estadística en un valor determinado. "Entity Statistics" generara estadísticas generales sobre la entidad, como información de costos tiempos y duración. "Time Interval" calculara y guardara la diferencia entre dos eventos. "Time Between" calculara la distancia en tiempo entre dos entidades que entran al módulos. "Expression" guardara el valor de la expresión especificada.

Nombre del Atributo (Attribute Name): nombre del atributo cuyo valor será usado para las estadísticas de intervalo. Solo se usara si el tipo es "Interval"

Valor (Value): valor que será registrado en la estadística cuando el tipo es "Expression" o añadido al contador cuando el tipo es "Count".

8.1.2. Módulos de Datos

Se visualizan el la hoja de calculo y son el conjunto de objetos que definen características de varios elementos del proceso, como recursos y colas.



Entidad Entity

Este módulos define los tipos de entidad y su imagen inicial en una simulación. Costos iniciales también se definen aquí.

Campos

Tipo (Type): el nombre de la entidad que esta siendo definida. Este nombre debe ser único.

Imagen inicial (Initial Picture): representación grafica de la entidad. Esta imagen puede ser cambiada durante la simulación, usando un módulos de asignación

Costo de Retención por hora (Holding Cost/Tour): costo por hora de procesar esa entidad en el sistema. Este costo se produce mientras la entidad este dentro de cualquier parte del sistema.

Valor Agregado Costo Inicial (Initial VA Cost): en realidad es un costo. Este costo se da mientras la entidad esta en una actividad que da valor agregado.

Valor No Agregado Costo Inicial (Initial NVA Cost): Este costo se da mientras la entidad esta en una actividad que **no** da valor agregado.

Costo de espera (Initial Waiting Cost): costo inicial en el que se incurra mientras la entidad esta esperando (como cuando se espera para completar un batch, o cuando se espera a que algún recurso este disponible)

Costo de Transferencia Inicial (Initial Transfer Cost): costo que implica transferir una entidad.

Otros Costos Iniciales (Initial Other Cost): costos en los que se incurren mientras la entidad esta en alguna otra actividad no contemplada en las opciones anteriores.

Reporta estadísticas (Report Statistics): especifica si se recolectaran estadísticas para este tipo de entidad.



Colas QUEUE

Este módulos se utiliza para especificar el tipo de cola, que por defecto es del tipo FIFO (First In – First Out). Primero que entra, primero que sale). Se utiliza para guardar la acumulación de trabajo para un recurso que esta en el módulos de proceso o como almacén para entidades que esperan para formar un batch.

Campos

Nombre (Name): nombre de la cola cuyas características están siendo definidas en este módulos, es un identificador único.

Tipo (Type): cola (FIFO), pila (LIFO), atributo mas bajo primero, atributo mas alto primero.

Atributo (Attribute Name): nombre del atributo que se evaluara para los tipos "Lowest attribute value" o "Highest Attribute value". Los empates se resolverán con la regla FIFO.

Compartido (Shared): checkbox que determina si una cola es usada en muchos lugares dentro de la simulación.

Reporta estadísticas (Report Statistics): especifica si se recolectaran estadísticas para esta cola.



Recursos Resource

Este módulo define los recursos en la simulación, incluyendo información de costos y disponibilidad. Los recursos pueden tener una capacidad fija que no varía durante la simulación, o depende de una agenda. Fallas y estados también pueden definirse en este módulo.

Este módulo sirve para representar todo tipo de recursos, como maquinarias, máquinas registradoras, líneas telefónicas o recursos humanos como recepcionistas, telefonistas u operadores.

Campos

Nombre (Name): nombre del recurso cuyas características están siendo definidas en este módulo, es un identificador único.

Tipo (Type): método para determinar la capacidad del recurso. "Fixed Capacity" (capacidad fija) no cambia durante la simulación. "Based On Schedule" (basado en horario), un módulo de agenda se usa para especificar la capacidad y la duración del recurso.

Capacidad (Capacity): número de unidades del recurso de un tipo que están disponibles para usar. Solo aplica si el tipo es Fixed Capacity.

Nombre de la agenda (Schedule Name): identifica el nombre de la agenda a ser usada por el recurso. Esta agenda definirá la capacidad del recurso de acuerdo al horario. Solo se utiliza si el tipo es Schedule.

Regla de Horario (Schedule Rule): dictamina cuando se cambia la capacidad de un recurso debido a una demanda excesiva. Aplica solo si el tipo es Schedule.

Costo de utilización por hora (Busy/Hour): costo por hora del recurso que está procesando una entidad. El recurso se pasa al estado ocupado cuando es asignado a una entidad y se "libera" cuando termina su trabajo. Durante el período ocupado, este costo es el que se acumula.

Costo de no uso por hora (Idle/Hour): costo de no-utilización del recurso, por hora. Durante el tiempo en que el recurso no está asignado a una entidad, este es el costo que se acumula.

Costo por uso (Per Use): costo del recurso, basado en su uso, independientemente del tiempo que le lleve realizar la tarea. Cada vez que se le asigna una entidad, se acumula este costo.

Conjunto de estados (State Set): conjunto de estados en los que el recurso puede encontrarse (ocupado, libre, anulado, en espera).

Estado Inicial (Initial State): estado inicial del recurso. Si se especifica, el nombre debe estar dentro del conjunto de estados. Este campo aparece solo si el conjunto de estados fue definido.

Fallas (Failures): lista todas las fallas que pueden asociarse al reporte.

Nombre de la falla (Failure Name): nombre de la falla asociada al recurso.

Regla de la falla (Failure Rule): comportamiento que debería ocurrir cuando una falla se encuentra en una unidad ocupada.

Reporta Estadísticas (Report Statistics): especifica si se recolectan o no estadísticas para este recurso.



Variables Variable

Este módulo se usa para definir variables, sus dimensiones y valores iniciales. Las variables se pueden referenciar en otros módulos, como el de decisión, cambiar sus valores con un módulo de asignación, o ser usadas en cualquier expresión. Sirven para registrar la cantidad de documentos procesados por hora, los números de serie que se asignan a partes para su identificación inequívoca, o el espacio disponible en un almacén.

Campos

Nombre (Name): nombre de la variable cuyas características están siendo definidas en este módulo, es un identificador único.

Filas (Rows) y Columnas (Columns): cantidad de filas y columnas de una variable dimensional.

Estadísticas (Statistics): checkbox que indica si se recolectan o no estadísticas para esta variable. Este campo solo está activo si los campos Rows y Columns están vacíos.

Borra todo (Clear Option): define la hora en la que el valor de la variable se resetea a su valor inicial. Si se especifica "Statistics", la variable se reseteará cuando las estadísticas se limpien. "System" indica resetear la variable cuando el sistema entero se reinicia. "None" indica que nunca se reseteará el valor, salvo al comienzo de la primera replicación.

Valores iniciales (Initial Values): valores iniciales de la variable.

Reporta Estadísticas (Report Statistics): especifica si se recolectan o no estadísticas para este recurso.



Agenda Schedule

En conjunto con el módulos de recursos, este módulos define una agenda de operación para el recurso. En conjunto con el módulos Crear, define los horarios de arribos de nuevas entidades. Puede utilizarse para modelizar los horarios de los empleados, incluyendo los descansos, tiempo invertido en el entrenamiento de nuevos empleados.

Campos

Nombre (Name): nombre de la variable cuyas características están siendo definidas en este módulos, es un identificador único.

Tipo de Formato (Format Type): formato de la agenda siendo definida. Si el formato es "Duration" (duración), el horario se define como un par de valores "valor-duración". Si el formato es "Calendar" (calendario), la agenda se define usando el editor de patrones de tiempo **Edit > Calendar Schedules > Time Patterns**.

Tipo (Type): tipo de agenda. Esta puede ser relacionada con la capacidad (capacity) en las agendas de recursos, relacionada con los arribos (arrival) cuando esta relacionada con el módulos Crear o puede ser de otro tipo (other) cuando se refiere a algún otro factor.

Unidad de tiempo (Time Units): unidad de tiempo usada para la información de duraciones.

Factor de Escala (Scale Factor): metodo para escalar la agenda de acuerdo a cambios en los valores de arribo u otros factores. El campo valor será multiplicado por este factor para determinar nuevos valores.

Duraciones (Durations): lista de pares valor-duración.

Valor (Value): representa la capacidad si el tipo es capacity, tasa de arribos si el tipo es arrival o algún otro tipo de valor si el tipo es other.

Duración (Duration Time): tiempo durante el cual el valor especificado en value es valido.



Conjuntos Set

En este módulos se definen conjuntos de ítems, como recursos, contadores, tipos de entidad o imágenes. Sirve para representar maquinas que hacen la misma tarea en una fabrica, vendedores o conjuntos de imágenes.

Campos

Nombre (Name): nombre del conjunto cuyas características están siendo definidas en este módulos, es un identificador único.

Tipo (Type): tipo de conjunto que esta siendo definido.

Miembros (Members): grupo de repetición que especifica el orden de los miembros del conjunto de recursos. Es importante cuando se usan reglas de selección como Preferred Order y Cyclical.

Nombre del recurso (Resource Name): nombre del recurso que será incluido en el conjunto. Solo aplica cuando el tipo es resource.

Contador (Counter Name): nombre del contador que será incluido en el conjunto. Solo valido cuando el tipo es counter.

Tipo de entidad (Entity Type): tipo de entidad que será incluido en el conjunto. Solo se usa cuando el tipo es entity.

Imagen (Picture Name): nombre de la imagen dentro del conjunto. Solo aplica si el tipo es entity picture.

8.2. Panel de Procesos Avanzados

En este capítulo se presentan los módulos del panel de Procesos Avanzados, con los cuales se puede agregar funcionalidad al modelado de los procesos. Se incluyen ejemplos de uso para cada módulo.

8.2.1. Los módulos de flujo

Los módulos de flujo son un conjunto de objetos, colocados en la ventana de diagrama para describir los procesos de simulación.



Módulo Retraso-Demora

Este módulo retrasa una entidad por una cantidad de tiempo específico. Cuando una entidad llega a el módulo de Retraso-Demora, la expresión "demora de tiempo" es evaluada y la entidad permanece en el módulo por el período de tiempo resultante. El tiempo es entonces adjudicado como valor agregado de la entidad, sin valor agregado, transferencia, en espera, o cualquier otra medida de tiempo. Los costos asociados son calculados y adjudicados.

Usos frecuentes

- Procesamiento de cheques en un banco
- Realizar el Setup de un equipo
- Transferencia de documentos a otros departamentos.

Campos

- Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.
- Adjudicación (Allocation): Tipo de categoría por la que la entidad incurre en tiempo de demora-retraso, con costos incluidos.
- Tiempo de demora-retraso (Delay Time): Determina la cantidad de retraso de la entidad.
- Unidades (Units): Unidades de tiempo utilizada para el tiempo de demora.

Módulo de Descenso-Reducción

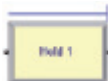
Este módulo remueve un número específico de entidades y las envía a otro módulo, tal como lo especifica la conexión gráfica. Deben ser definidas por el usuario atributos de valores y atributos internos para el descenso-reducción de entidades, basados en reglas puntuales.

Usos frecuentes

- Cargar estanterías con productos.
- Separar un formulario para el uso de varios departamentos.

Campos

- Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.
- Cantidad (Quantity): es el número de entidades que serán reducidas del grupo.
- Rango de salida (Starting Rank): Basado en las entidades del grupo, son aquellas que serán reducidas.
- Atributos de los Miembros (Member Attribute): método para determinar cómo se asignarán los valores representativos (diferentes de costos y tiempos), de las entidades a ser reducidas.
- Nombre del Atributo (Attribute Name): Nombre representativo del atributo asignado para las entidades a ser reducidas del grupo original.

**Módulo de Espera**

Este módulo retendrá una entidad en cola a la espera de una señal, a la espera de que una condición específica se convierta en realidad, o quede en espera indefinidamente (siendo eliminada más tarde con el Módulo de Eliminación (Remove module)).

Si la entidad espera por una señal, el Módulo-Señal es utilizado desde cualquier lugar para permitir a la entidad pasar al módulo siguiente. Si la entidad espera por una condición dada a que se convierta en verdadera, permanecerá en el módulo (tanto en una cola definida como en una interna) hasta que las condiciones se cumplan. Cuando la entidad está en una espera infinita, el Módulo de Eliminación (Remove module) es utilizado desde cualquier etapa para permitir que la entidad continúe el proceso.

Usos frecuentes

- Espera por una luz de tráfico para que pase a verde
- Retener una parte a la espera de autorización.
- Chequeo del estado de una máquina u operador para continuar un proceso.

Campos

- Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.
- Type (Tipo): Indica las razones para detener una entidad en una cola específica o interna. Esperar por la Señal (Signal) retendrá la entidad hasta que una señal de igual valor sea recibida. La búsqueda de la condición (Scan for Condition) retendrá la entidad hasta que la condición se convierta en verdadera. Espera Infinita (Infinite Hold) retendrá la entidad hasta que ésta sea removida de la cola con el Módulo Remover (Remove Module).
- Esperar por Valor (Wait for Value): Código para la espera de la entidad. Aplica únicamente cuando el Tipo (Type) es Esperar por Valor (Wait for Signal).
- Límite (Limit): Número máximo de entidades a ser liberadas una vez recibida una señal. Aplica sólo cuando el Tipo (Type) es Esperar por Señal (Wait for Signal).

Condición: (Condition): Especifica la condición que será evaluada para retener la entidad en el módulo. Si la condición resulta verdadera, la entidad abandona el módulo inmediatamente. Si la condición es falsa, la entidad aguardará en la cola asociada hasta que la condición se vuelva verdadera. Aplica únicamente cuando el Tipo (Type) es Búsqueda de Condición (Scan Condition).

Cola Tipo (Queue Type): Determina el tipo de colas utilizadas para retener las entidades. Si la Cola es seleccionada, el nombre de la cola es especificado. Si el conjunto es seleccionado, el conjunto de colas y sus miembros del conjunto serán especificados. Si Interna es seleccionada, la cola Interna es utilizada para retener a todas las entidades en espera. Atributos y Expresiones son métodos adicionales para definir las colas que serán utilizadas.

Nombre de Cola (Queue Name): Este campo es visible únicamente si la Cola Tipo (Queue Type) es Cola (Queue), y define el símbolo nombre de las colas.

Conjunto de Nombres (Set Name): Este campo es visible únicamente si la Cola Tipo es Conjunto (Queue Type is Set), y define el conjunto de colas que contienen las colas sean referenciadas.

Índice de Nombres (Set Index): Este campo es visible únicamente si la Cola Tipo es Grupo, e identifica el índice dentro del conjunto de la cola. Nótese que es el índice dentro del Conjunto y no el nombre de la Cola en el Conjunto. Por ejemplo, la única entrada válida para el conjunto Cola que contiene tres miembros es la expresión que evalúa a 1, 2 o 3.

Atributo (Attribute): Este campo es sólo visible si la Cola Tipo es Atributo (Attribute). El atributo ingresado en este campo será evaluado para indicar cuál cola debe ser utilizada.

Expresión (Expression): Este campo es sólo visible si la Cola Tipo es Expresión. La expresión ingresada en este campo será evaluada para indicar cuál cola debe ser utilizada.



Módulo Juego (Match)

Este módulo tiene un número específico de entidades esperando en diferentes colas. El Juego se realizará cuando halla al menos una entidad en cada una de las colas deseadas. Adicionalmente se debe especificar un atributo para que las entidades en espera en las colas tengan los mismos valores de atributos antes que el Juego se inicie.

Cuando una entidad ingresa al Módulo Juego, será colocada en una de las hasta cinco colas asociadas, basado en el punto de entrada a la que están conectadas. Las entidades permanecerán en sus colas respectivas mientras exista el Juego.

Una vez iniciado el Juego, una entidad de cada una de las colas será liberada para ser combinada. Las entidades combinadas son sincronizadas para salir del módulo.

Usos frecuentes

Montaje de una parte.

Reunir varios productos para armar una orden.

Sincronizar la salida de un cliente con una orden completa.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Número a Combinar (Number to Match): Cantidad de entidades combinables que deben alojarse en distintas colas antes de que las combinaciones se completen.

Tipo (Type): Método de combinación de las entidades entrantes. Si "Tipo" es Alguna Entidad, debe haber una entidad en cada cola para que la combinación sea hecha. Si el Tipo está Basado en el Atributo, debe haber una entidad en cada cola con el mismo valor atributo.

Nombre de Atributo (Attribute Name): El nombre del Atributo es utilizado para identificar el valor de combinación de la entidad entrante.



Módulo Pickup

Este módulo remueve un número consecutivo de entidades de una cola dada, comenzando en un rango específico de la misma. Las entidades removidas son agregadas al final de grupo.

Usos frecuentes

Reunir en una orden varias colas.

Reunir formularios completos para una orden.

Recoger estudiantes de una parada de ómnibus para ir a la escuela.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Cantidades (Quantities): Cantidad de entidades a ser tomadas.

Nombre de Cola (Queue Name): Nombre de la cola desde donde las entidades serán seleccionadas, comenzando en un rango específico.

Comienzo de Rango (Starting Rank): Rango de comienzo de las entidades a ser seleccionadas de la cola.



Módulo de Lectura/Escritura

Este módulo es utilizado para leer información de un archivo de entrada o de un teclado y asignar datos de valor a una lista de variables o atributos (o cualquier otra expresión). Este módulo es también utilizado para escribir información en un dispositivo de salida, tanto a pantalla como un archivo.

Tanto al leer desde o escribir hacia un archivo, el módulo de Lectura/Escritura varía según el Tipo de Nombre de Archivo Arena.

Archivo Secuencial o Hoja de Cálculo de LOTUS. Cuando una entidad ingresa al módulo Lectura/Escritura, el archivo específico es examinado para ver si está activo. De no estarlo, es automáticamente abierto. Los valores de los atributos, variables o expresiones son leídos o grabados de acuerdo a lo especificado.

Microsoft Excel, Microsoft Access, y ActiveX Data Objects Access Types.

Cuando una entidad ingresa al Módulo de Lectura/Escritura el archivo específico es examinado para chequear si está abierta (activa) una conexión ADO al archivo. Si la tiene cerrada, una conexión es automáticamente abierta hacia la fuente de información utilizando ya sea Microsoft Jet OLE (si el tipo de acceso es Excel o Access) o la cadena de conexión específica (si el tipo de acceso es ActiveX Data Objects).

La estructura de datos (Recordset ID) es entonces examinada, para ver si ese recordset en el archivo está abierto. Un estructura de datos (recordset) se refiere a una tabla de datos, en el almacenamiento de datos, que esta organizada en columnas (campos) y filas (registros).-

Si la estructura de datos esta cerrada, entonces esta es automáticamente abierta utilizando la definición especificada en el módulos de archivo de la estructura de datos. (ejemplo: el rango de nombres especificado o el nombre de la tabla).

La estructura de datos permanece abierta hasta que el archivo es cerrado.

Los valores de atributos, variables o expresiones, listados en la asignación de grupos repetitivos son leídos desde o grabados en el Número de Registro Especificado, en la estructura de datos. La primer entrada en las asignaciones (Assignments) es leída desde o grabada en el primer campo en el registro, la segunda entrada corresponde al segundo campo y así sucesivamente.

Extensible Markup Language Access Type.

Cuando una entidad ingresa al Módulo de Lectura/Escritura, el archivo es examinado para ver si está activo. Si no lo está, será automáticamente abierto en una planilla de datos ADO.

Los valores de los atributos, variables y expresiones listados son leídos desde o grabados en el Número de Registro específico, en la planilla de datos.

Usos frecuentes

Lectura en línea de arribos/partidas de información.

Preparar un usuario final para el modelo de configuración de datos.

Guardar información en un archivo de datos, como ordenes de arribo, tiempos, tiempos de terminación.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Tipo (Type): Método de lectura o escritura utilizado. La información debe ser leída desde un archivo o desde el teclado. La información debe ser enviada a un archivo o a la pantalla.

Nombre de Archivo Arena (Arena File Name): Nombre utilizado para identificar el archivo en el Módulo Archivo.

Anular Formato de Archivos (Overriding File Format): Formato para guardar o leer información de un archivo. Este formato anulará cualquier formato especificado en la estructura del archivo o en el Módulo Archivo. A FORTRAN o un formateo del disco C:, puede ser utilizado para describir el tipo y localización de cada archivo.

Identificación de Acceso al conjunto de datos (Recordset ID Access Type): Identifica el conjunto de datos de los archivos Arena a los cuales se va a acceder. Aplica únicamente cuando son archivos Microsoft Excel, Access, o ActiveX Data Objects.

Número de Registro de Acceso (Record Number Access Type): Número de registro (ej. Fila) en la identificación de conjunto de datos para leer desde o grabar en. Aplica únicamente en archivos Microsoft Excel, Access, ActiveX Data Objects, o Extensible Markup Language.

Tipo Variable (Variable Type): Tipo de información que será leída o grabada.

Nombre de Atributo (Attribute Name): Define el símbolo de nombre del atributo a ser grabado o leído.

Nombre de Variable (Variable Name): Define el símbolo de nombre de la variable a ser grabada o leído.

Fila (Row): Especifica el índice fila para un grupo de variables.

Columna (Column): Especifica el índice columna para un grupo de variables.

Otros (Other): Define la expresión de lectura o escritura de otro tipo de información.



Módulo de Liberación

Este módulo se utiliza para liberar unidades de un recurso que una entidad ha aprovechado previamente. Este módulo puede ser utilizado para liberar recursos individuales o en forma conjunta. Para cada recurso liberado, el nombre y la cantidad debe ser especificada.

Cuando una entidad ingresa al Módulo de Liberación, deja el control del recurso. Cualquier entidad esperando en cola por esos recursos ganará control sobre ellos inmediatamente.

Usos frecuentes

Finalización de la orden de un cliente (liberación del operador)

Finalizar la devolución de impuestos (liberación del contador)

Alta de un Hospital (liberación de doctor, enfermera, habitación)

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Tipo (Type): Tipo de recurso a ser liberado, ya sea especificando un recurso particular o seleccionando el mismo de un pool de recursos. El nombre del recurso puede ser especificado también por una expresión o un valor de atributo.

Nombre del Recurso (Resource Name): Nombre del recurso que será liberado.

Nombre de Conjunto (Set Name): Nombre del conjunto de recursos del cual un miembro va a ser liberado.

Nombre de Atributo (Attribute Name): Especifica el nombre del recurso a ser liberado.

Expresión (Expression): Especifica el nombre del recurso a ser liberado.

Cantidad (Quantity): Cantidad de recursos con un nombre o de un conjunto que serán liberados. Para los conjuntos, este valor indica sólo el número del recurso seleccionado no la cantidad de miembros a ser liberados del conjunto.

Regla de Liberación (Release Rule): Método para determinar qué recurso de un conjunto será liberado. El último miembro aprovechado y el primer miembro aprovechado serán liberados primero/último desde el conjunto donde fue aprovechado. El número específico indica que un número de miembro o atributo será usado para especificar el miembro a ser liberado.

Índice de Conjunto (Set Index): Índice del conjunto de recursos que la entidad va a liberar.



Módulo de Eliminación

Remueve una sola entidad desde una posición específica en la cola y la envía al módulo designado.

Cuando una entidad ingresa al Módulo de Eliminación, la elimina de la cola y la envía al módulo conectado. El rango de es la ubicación de la entidad en la cola. La entidad que causa la eliminación pasa al siguiente módulo especificado y será procesado antes que la entidad eliminada

Usos frecuentes

Remover una orden de la cola que debe completarse a continuación.

Llamar una paciente de la sala de espera para ser examinado.

Recuperar la siguiente orden a ser procesada de una lista de documentos.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Nombre de la cola (Queue Name): Nombre de la cola desde la cual la entidad será removida.-

Rango de la entidad (Rank of Entity): Rango de la entidad que será removida de la cola.



Módulo Seize

El módulo Seize asigna unidades de uno o más recursos a una entidad. Este módulo puede ser usado para tomar unidades de un recurso en particular, un miembro de un conjunto de recursos, o un recurso definido por un recurso alternativo, como un atributo o expresión.-

Cuando una entidad ingresa al módulo, esta espera en cola (si esta especificado) hasta que todos los recursos estén disponibles simultáneamente.

Usos frecuentes

Comenzar una orden de un cliente

Comenzar una devolución de impuestos.

Ser admitido en un hospital.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Asignación (Allocation): determina a qué categoría de costo de uso de recursos será asignada a una entidad que irá hacia el Módulos de Aprovechamiento.

Prioridad (Priority): Valora la entidad en espera en este módulo para el recurso especificado, si una o más entidades de otros módulos están esperando el mismo recurso.

Tipo (Type): Tipo de recurso a ser aprovechado, ya sea especificando un recurso en particular, o seleccionando el mismo de un pool de recursos. El nombre del recurso puede ser especificado como un valor de atributo o dentro de una expresión.

Nombre de Recurso (Resource Name): Nombre del recurso a ser utilizado.

Nombre de Conjunto (Set Name): Nombre del conjunto de recursos del cual un miembro será utilizado.

Nombre de Atributo (Attribute Name): Nombre del atributo que almacena el nombre del recurso a ser utilizado.

Expresión (Expression): Evalúa el nombre de un recurso a ser utilizado.

Cantidad (Quantity): Número de recursos de un nombre dado por un conjunto a ser aprovechado. Para conjuntos, este valor especifica únicamente el número del recurso seleccionado, no la cantidad de miembros.

Regla de selección (Selection Rule): Método de selección de los recursos disponibles de un conjunto. En forma cíclica, lo hará cíclicamente sobre los miembros disponibles. De manera aleatoria, lo hará aleatoriamente. Con orden preferencial, siempre seleccionará el primer miembro disponible. Miembro específico, requiere un valor de atributo de entrada para especificar el miembro del conjunto.

Salvar Atributo (Save Attribute): Se utiliza para almacenar el número de índice del conjunto de miembros elegidos. Este atributo puede luego ser referenciado con la regla de Selección Específica de Miembro.

Conjunto de Índices (Set Index): Valor dentro del conjunto que identifica al miembro solicitado. Si un nombre de atributo es utilizado, la entidad debe tener un valor de atributo antes de utilizar esta opción.

Estado del Recurso (Resource State): Estado del recurso que será asignado después que el recurso sea utilizado. El estado del recurso debe ser definido con el Módulos de Recurso.

Tipo de Cola (Queue Type): Determina el tipo de cola utilizada para retener las entidades mientras esperan para aprovechar un recurso o recursos. Si la cola es seleccionada, el nombre de la cola es especificada. Si el conjunto es seleccionada, el conjunto de cola y el miembro en el conjunto son especificados. Si Interna es seleccionada, una cola Interna es usada para retener todas las entidades. Atributos y expresiones son métodos adicionales para definir las colas que serán utilizadas.

Nombre de Cola (Queue Name): Es visible únicamente si la Cola Tipo es Cola.

Nombre del Conjunto (Set Name): Visible únicamente si el Tipo Cola es Conjunto y define el conjunto de Colas que contiene la cola en referencia.

Conjunto Índice (Set Index): Define el índice dentro del conjunto de cola. Nótese que es el índice dentro del conjunto y no el nombre del conjunto de cola. Por ejemplo, la única entrada válida para una cola que contiene tres miembros es una expresión que evalúe 1, 2 ó 3.

Atributo (Attribute): El atributo de este campo será evaluado para indicar qué cola será usada.

Expresión (Expression): La expresión ingresada en este campo será evaluada para indicar qué cola será utilizada.



Módulo de Búsqueda

Este módulo busca una cola, un grupo, o una expresión para encontrar el rango de entidad, o el valor de rango global de la variable "J" que satisface las condiciones específicas de búsqueda. Si buscando una cola o un grupo, el valor variable del sistema global "J" es colocado en el rango de la primer entidad que cumpla con las Condiciones de Búsqueda, o con cero si las Condiciones de Búsqueda no se cumplen.

Cuando se busca una expresión, el sistema global variable "J" es agregado al valor del primer índice que cumple con las condiciones de búsqueda o con cero si ningún valor de "J" en el rango especificado satisface las condiciones de búsqueda.

Cuando una entidad ingresa al Módulo de Búsqueda, el índice "J" es agregado al comienzo del índice y entonces las condiciones de búsqueda son chequeadas. Si las condiciones de búsqueda son satisfactorias la búsqueda finaliza y el valor "J" es retenido. O de lo contrario, el valor de "J" puede incrementarse o disminuir y la condición es re-chequeada. El proceso se repite hasta que las condiciones de búsqueda sean satisfactorias o el valor final re-chequeado.

Si la condición no coincide o no hay entidades en la cola o grupo, "J" será puesto como igual a cero.

Usos frecuentes

Búsqueda de un número de orden en particular de una cola.

Búsqueda en un grupo por una parte específica.

Determinar qué proceso ingresar basado en la disponibilidad de recursos.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Tipo (Type): Determina qué va a ser buscado. Las opciones de búsqueda incluyen entidades de una cola, entidades de un grupo o alguna expresión.

Nombre de Cola (Queue Name): Nombre que será buscado. Aplica sólo cuando el tipo busca una cola.

Valor de Comienzo (Starting Value): Rango de partida en una cola o grupo o valor de inicio para "J" en una expresión.

Valor de Finalización (Ending Value): Fin de rango en una cola o grupo o valor final para "J" en una expresión.

Condición de Búsqueda (Search Condition): Condición que contiene el índice "J" para buscar expresiones o contienen el nombre de atributos para la búsqueda de colas.



Módulo de Señal

Este Módulo envía valores de señal para cada módulo de Espera, que espera la señal para liberar el máximo de entidades especificadas.

Cuando una entidad ingresa al Módulo Señal, la señal es evaluada y el código de la señal es enviada. Al mismo tiempo, entidades del Módulo de Espera que están esperando por la misma señal son removidas de sus colas.

La entidad que envía la señal continúa el proceso hasta que encuentra un retraso, ingresa en cola, o sea dispuesta.

Usos frecuentes

Analizar patrones de tráfico en una intersección.

Avisar a un operador que complete una orden para la que esperaba un componente.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Valor de Señal (Signal Value): Valor de señal que debe ser enviado a las entidades en los módulos de espera.

Límite (Limit): Número máximo de entidades que deben ser liberadas del Módulo de Espera cuando la señal es recibida.



Módulo de Almacenamiento

Este Módulo agrega una entidad al almacenamiento de datos. El Módulo "Un-Store" será utilizado entonces para eliminar una entidad del almacenamiento.

Cuando una entidad ingresa a este módulo, ese almacenamiento específico se incrementa y la entidad pasa inmediatamente al siguiente módulo en el modelo.

El almacenamiento es útil para desplegar entidades animadas mientras una entidad atraviesa procesos en otros módulos. Además, las estadísticas se toman del número de entidades almacenadas.

Usos frecuentes

Rastreo del número de clientes en un supermercado.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Nombre del Almacenaje (Storage Name): Nombre por el cuál una entidad será agregada.

Nombre del Conjunto (Set Name): Nombre del conjunto de Almacenaje por el cual el almacenamiento será seleccionado.

Conjunto de Índice (Set Index): Índice dentro del conjunto de almacenaje que contiene el nombre del almacenaje deseado.

Atributo (Attribute): Nombre del atributo del valor que contiene el almacenaje.

Expresión (Expression): Expresión que es evaluada para el almacenaje donde una entidad es colocada.



Módulo de Eliminación del Almacenaje (Un-Store)

Remueve la entidad del Almacenaje. Cuando una entidad ingresa a este módulo, el almacenaje especificado disminuye y la entidad inmediatamente pasas al siguiente módulo del modelo.

Usos frecuentes

Eliminar una entidad cuando un proceso se completa.

Rastrear el número de clientes en un supermercado.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

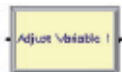
Nombre del Almacenaje (Storage Name): Nombre por el cuál una entidad será agregada.

Nombre del Conjunto (Set Name): Nombre del conjunto de Almacenaje por el cual el almacenamiento será seleccionado.

Conjunto de Índice (Set Index): Índice dentro del conjunto de almacenaje que contiene el nombre del almacenaje deseado.

Atributo (Attribute): Nombre del atributo del valor que contiene el almacenaje.

Expresión (Expression): Expresión que es evaluada para el almacenaje donde una entidad es colocada.



Módulos de Ajuste de Variables

Este módulo ajusta una variable a un valor de target de un rango específico.

Este módulo debe ser utilizado con la opción "Rate by Expression" en Imágenes Globales y Recursos para presentar animación rotativa de las imágenes.

Puede ser utilizada también para aproximar/animar un continuo incremento o disminución del valor de una variable en el tiempo.

Cuando una entidad ingresa a este Módulo, la Variable Nombre es ajustada al Rango de Valores especificados. La entidad permanece en el módulo hasta que los ajustes se finalicen.

La Actualización del Intervalo (Update Interval) especifica el tiempo del intervalo entre la asignación actual del valor variable sobre el período de tiempo del cambio.

Una actualización de intervalo menor produce una animación en planos e imágenes rotativas que referencian la variable, así como dan más precisión en el recabado de información estadística.

Una gran cantidad de actualizaciones de intervalos produce una aceleración en la velocidad del proceso.

Usos frecuentes

Rotar una imagen 180 grados con un rango de 5 grados por segundo.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Tipo de Variable (Variable Type): Especifica el conjunto de características de la variable a ser ajustada.

Fila (Row): Especifica la fila en el índice.

Columna (Column): Especifica la columna en el índice.

Nombre de la Variable (Variable Name): Especifica el nombre de la variable a ser ajustada.
 Valor (Value): Especifica el target del valor al cuál la variable va a ser ajustada.
 Rango (Rate): Especifica el rango en el cual se harán los ajustes.
 Unidad (Units): son las unidades de tiempo del rango.
 Intervalos de actualización (Update Interval): Especifica el intervalo entre actualizaciones de la variable.
 Adjudicación (Allocation): Tipo de categoría por la cuál la entidad tiene una demora, tiempos y costos en el módulo serán agregados.

8.2.2. Módulos de Datos

Son conjuntos de objetos de una hoja de cálculo del modelo que define las características de varios elementos de procesos, como colas y recursos.

Módulo de Conjuntos Avanzados

Este módulo especifica conjuntos de colas, conjuntos de almacenaje, y otros conjuntos con sus respectivos miembros. Un conjunto define un grupo de elementos similares que pueden ser referenciados por un nombre común y luego ingresado al índice. Los elementos que componen un conjunto son referenciados para nosotros como miembros del conjunto.

El conjunto de colas pueden ser especificadas dentro del módulo de aprovechamiento/ disponibilidad de materiales. Conjuntos de almacenaje pueden ser usados en los módulos de Storage y Un-Storage.

Usos frecuentes

Varios puntos de salida en un supermercado

Módulo Expresión

Define expresiones y éstas son asociadas a valores. Las expresiones son referenciadas en el modelo por su nombre.

El valor de una expresión puede estar formado por una combinación de integrantes, números reales, símbolos, nombres, distribuciones estáticas (ej. NORM(10,2)), operaciones aritméticas, paréntesis, operadores lógicos, atributos y variables.

Un valor de expresión puede estar referenciada dentro de otra expresión, pero llamados recursivos a la misma expresión no están permitidos.

Usos frecuentes

Distribución de una entidad que ingresa al módulo.

Valores de imagen de una entidad.

Expresión compleja para tiempo de orden de entrada.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Fila (Row): Número máximo de filas en una expresión definida.

Columna (Column): Número máximo de Columna en una expresión definida.

Módulo de Fallo

Este módulo está designado para el uso con recursos. Cuando ocurre una falla, el recurso entero falla. El módulo está designado para ser utilizado con capacidad de recursos individuales o múltiples.

Usos frecuentes

Caída de la información de un equipo.

Recarga de cintas de una caja registradora, cada "X" cantidad de clientes.

Apagados aleatorios de equipos.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Tipo (Type): Determina si la falla es determinada en base a tiempo o a una cuenta.

Cuenta (Count): Determina la cantidad de recursos liberados.

Tiempo disponible (Up Time): Define el tiempo entre fallas, para fallas basadas en tiempo

Unidades de tiempo disponible (Up Time Units): Unidades de tiempo entre fallas

Tiempo de Falla (Down Time): Define la duración de la falla.

Unidades de Tiempo de la Falla (Down Time Units): Duración de unidades de tiempo de la falla.

Tiempo Disponible en este estado únicamente (Uptime in this State only): Define el estado que debe ser considerado para el tiempo entre fallas. Si el estado no está especificado, entonces todos los estados serán considerados. Por ejemplo, usted puede querer definir una falla basada únicamente en el estado "Ocupado", y por lo tanto, el tiempo entre fallas debe estar basado en la cantidad de tiempo que el recurso está ocupado, sin simular tiempo reloj.

Módulo Archivo

Este módulo debe ser incluido cada vez que se accede a un archivo externo utilizando el Módulo de Lectura/Grabado. Este módulo identifica el nombre del archivo del sistema y define el método de acceso al mismo, formatos y características funcionales del archivo.

Usos frecuentes

Archivos que contienen información pre definida de vuelos.

Archivos con información específica de órdenes de clientes.

Campos

Nombre (Name): identificador único que se encuentra dentro del símbolo del módulo.

Nombre del Archivo de Sistema Operativo(Operating System File Name): Nombre de archivo actual que está siendo leído y al cuál será grabada información

Acción de Fin de Archivo(End of File Action): Tipo de acción que ocurre si la condición de fin de archivo es alcanzado.

Opción de Inicialización (Initialize Option): Acción que debe tomarse sobre un archivo al comienzo de cada proceso de simulación.

Comentario (Comment): Carácter que indica un registro de comentario.

Conjunto de Nombres Grabados (Recordset Name): Este nombre debe ser único por archivo.

Módulo de Conjunto de Estados

Este módulo es utilizado para definir el estado de los recursos. Los estados deben estar asociados. El módulo de Recursos en el Panel de Procesos Básicos referencia el conjunto de estados, los cuáles son utilizados para los recursos.

Usos frecuentes

Estados para un operador como por ejemplo: ocupado con un cliente, ocupado reponiendo estanterías, ocupado al teléfono.

Estados de un equipo, como con falla aleatoria, ocupado por el setup, ocupado en procesos.

Estados de un doctor, como agendando citas, en ronda Hospitalaria, en papeleo, con llamadas telefónicas.

Campos

Nombre del conjunto de Estados (StateSet Name): Conjunto de estados de recursos que deben ser asignados durante una simulación.

AutoEstado o Falla (AutoState or Failure): Usado para asociar el nombre estado a un auto estado (como ocupado, inactivo, o Falla) o un nombre particular de Falla. Si no es utilizado, el estado debe asignarse ya sea al Módulo de Asignación o al Estado Recurso.

Módulo Estadístico

Este módulo es utilizado para definir estadísticas adicionales que deben ser realizadas durante la simulación y deben especificar el archivo de salida de datos. Mientras que el resultado estadístico es automáticamente generado, si se especifica un archivo de salida, entonces cada observación individual es grabada en ese archivo.

Usos frecuentes

Recolectar datos de observación para comparar dos configuraciones simuladas

Colectar información estadística adicional como cantidad de colas y cantidad de recursos ocupados

Generar estadísticas de los diferentes estados de los operarios o de los equipos.

Campos

Tipo (Type): Tipo de estadística a ser definida.

Nombre Contador (Counter Name): Aparece en el reporte como identificador del contador.

Tipo de frecuencia (Frequency Type): Indica si las estadísticas están siendo recolectadas de un recurso o de una expresión

Etiqueta de Reporte (Report Label): Este nombre aparece como etiqueta de reporte asociado con la expresión en el reporte. Si ya estadística ya ha sido definida a través de otro módulo, el archivo de etiqueta de reporte aparecerá automáticamente.

Límite (Limit): Se utiliza para definir la condición de finalización para una simulación. Cuando el contador alcanza o excede el límite especificado, la simulación termina y el reporte es generado.

Opción de Comienzo (Initialization Option): Indica cuando el contador debe comenzar. Si el archivo es especificado como "NO" y las múltiples duplicaciones son hechas, entonces el valor del contador al final del proceso será retenido como el valor de inicio al comienzo del siguiente proceso. Cuando es especificado con "SI", el valor del contador es seteado en cero al comienzo de cada proceso. Si se setea para Repetir, el valor del contador será borrado cuando otras estadísticas lo hagan, como lo especifica el setup.

Rango Constante (Constant Range): Especifica cuando la estadística está siendo realizada en base a Valore de Expresión como un Valor Constante o como un Rango de Valores.

Valor (Value): Especifica el valor cuando la recolección estadística es un valor constante. Por lo tanto, la estadística será recolectada en esta categoría cuando la Expresión Valor equipare el Valor. Si en la recolección estadística de un valor de Rango, el archivo especifica el valor más bajo en el rango, las estadísticas serán recolectadas sobre esta categoría, más allá que le Valor Expresión sea más grande que Valor y menor que o igual al Valor.

Valor Alto (High Value): Especifica el mayor valor.

Nombre de Categoría (Category Name): Este nombre aparece en el sumario como identificador de categoría.

**Módulo de Almacenamiento**

Los almacenamientos son automáticamente creados por cualquier módulo que haga referencia el almacenaje, por lo cuál este módulo raramente es utilizado. En el único momento que se utiliza es cuando un almacenamiento es definido como miembro de un conjunto o especificado por un atributo o expresión.

Campos

Nombre : (Name): El nombre por el cual el almacenamiento va a ser definido. Este nombre debe ser único.

9. Anexo B: Diagrama de flujos del modelo

10. Anexo C: Reporte de Arena

11. Bibliografía

- ACI (Airports Council Internacional) www.aci.aero
- Arena User's Guide, Rockwell Automation (2007)
- Arena online help. Contenido del menú de ayuda de Arena
- Doshi, N., Moriyama, R.: "Application of simulation models in airport facility design", Proceedings of the 2002 Winter Simulation Conference
- Gatersleben, M. R., van der Weij, S. W.: "Analysis and simulation of passenger flows in an airport terminal", Proceedings of the 1999 Winter Simulation Conference
- International Air Transport Association: "Airport development reference manual. Volume I", 9na. edición (2004)
- Janic, M.: "Terminal sizing theory", Loughborough University (2000)
- Joustra, P. E., Van Dijk, N. M.: "Simulation of check-in at airports", Proceedings of the 2001 Winter Simulation Conference
- Kelton, D., Sadowsky, R., Sturrock, D.: "Simulation with Arena", 4ta. edición, Mc Graw Hill (2007)

- Kiran, A. S., Cetinkaya, T., Og, S.: "Simulation modeling and análisis of a new international terminal", Proceedings of the 2000 Winter Simulation Conference
- Law, A.: "Simulation modeling and analisis", 4ta. edición, Mc Graw Hill (2007)
- Nakayama, M. K.: "Output analysis for simulation", Proceedings of the 2006 Winter Simulation Conference
- Pendergraft, D. R., Robertson, C. V., Shrader, S.: "Simulation o fan airport passenger security system", Proceedings of the 2004 Winter Simulation Conference
- Schumacher B., Lalsare, R.: "Simulation at Delta Airlines"
- Snowdon, J. L., Montevecchi, M., MacNair, E., Callery, C. A.: "Ending the blues for airline travelers", Simulations- und Automations- Technologie GMBH, IBM Global Services (1998)
- Takakuwa, S., Oyama, T.: "Simulation análisis of international-departure passeger flows in an airport terminal", Proceedings of the 2003 Winter Simulation Conference
- Thesen, A., Travis, L.: "Introduction to simulation", Proceedings of the 1991 Winter Simulation Conference
- Verbraeck, A., Valentin, E.: "Simulation building blocks for airport terminal modeling", Proceedings of the 2002 Winter Simulation Conference

