



Flexsim, una nueva alternativa para la ingeniería hacia la toma de decisiones en la operación de un sistema de múltiples estaciones de prueba.

Marco Antonio Díaz Martínez, Alma Cruz Méndez, Jesús Gómez Castellanos

Instituto Tecnológico Superior de Pánuco

marco.diaz@itspanuco.edu.mx

RESUMEN

El objetivo principal de este documento es afrontar conceptos clave en relación con el uso del software FlexSim dentro de un modelo de simulación, como una alternativa innovadora de efectividad en la toma de decisiones de las operaciones de un sistema de producción. El estudio incluye una revisión literaria, conceptos del modelo y simulación, aplicaciones, la caracterización de Flexsim y la descripción para la construcción de un modelo en el software FlexSim. Este trabajo se desarrolla partiendo del entendimiento y uso del software, modelación, ejecución y análisis de las operaciones de un sistema de múltiples estaciones de prueba. El modelo aquí descrito ilustra claramente la amplia precisión y capacidad de FlexSim como herramienta de mejora, además muestra lo sencillo que es su programación y su estilo innovador en el diseño en 3D. Este artículo busca dar una idea clara del impacto que tiene la simulación dentro de un sistema de producción y a su vez logre un entendimiento claro del uso de Flexsim.

INTRODUCCIÓN

Para poder comprender la realidad y la dificultad que un sistema puede conllevar, ha sido necesario construir un modelo artificialmente con objetos y experimentar con ellos dinámicamente antes de interactuar con el sistema real. La simulación por computadora puede observarse como el equivalente computarizado a ese tipo de experimentación. Para lo cual es necesario construir modelos que representen la realidad de tales modelos que puedan ser interpretados por una computadora. Las operaciones, procesos o servicios de varias clases que existen en el mundo real son generalmente llamados sistemas y para estudiarlos de forma científica frecuentemente se hace un conjunto de suposiciones acerca de cómo éstos trabajan. Estas suposiciones, que generalmente

toman la forma de relaciones matemáticas o lógicas constituyen un modelo que es usado para intentar ganar un entendimiento de cómo se comporta el sistema correspondiente. Si las relaciones entre los componentes que componen el modelo son bastante simples, puede ser posible usar métodos matemáticos (tales como, cálculo o estadística) para obtener información exacta en cuestiones de interés, a esto se le llama solución analítica. Sin embargo, la mayoría de los sistemas reales son difíciles, por lo que deben ser estudiados por medio de simulación para permitir que estos modelos sean reales y puedan ser evaluados de forma analítica. Las áreas de aplicación de la simulación son numerosas y diversas; es usual encontrar aplicaciones en ingeniería, economía, manufactura, administración, ciencias sociales.

El mostrar un ambiente concreto de simulación desde una perspectiva aleatoria en modelos prácticos utilizando el software FlexSim, requiere del entendimiento de los conceptos elementales que componen esta relación. En este contexto, es necesario tener una buena comprensión teórica de los conceptos de modelado y simulación de procesos industriales. Además de contar con conocimientos en el desarrollo de análisis experimental, en esta investigación se describen los conceptos principales del tema en estudio.

El objetivo de este artículo es inducir el uso de la simulación en la evaluación de un sistema productivo en un ambiente virtual, con la intención de lograr la máxima eficiencia en los sistemas de producción, minimizar costos, mejorar la calidad, reducir el lapso entre la fabricación y la entrega de los productos al cliente, proyectar escenarios catastróficos y extremos. Así como la aplicación de alternativas educativas para la formación de personal en diferentes áreas de una organización.

ANTECEDENTES

Definición de simulación

Para el desarrollo del presente trabajo resulta de suma importancia iniciar definiendo el término simulación. La simulación es la representación gráfica de algún proceso, sistema de producción o sistema real a través del tiempo, ya sea diseñada manualmente o en una computadora, la simulación compromete la generación de un historial artificial de un sistema y su observación para obtener inferencias relacionadas con las características

operativas del sistema real (Banks, Carson, Nicol, & Nicol, 2005). La simulación por computadora se refiere a los métodos para estudiar una gran variedad de modelos de sistemas del mundo real mediante la evolución numérica usando un software diseñado para imitar las características u operaciones del sistema, a menudo en el transcurso del tiempo (Kelton, Sadowski, & Sturrock, 2008).

La simulación comienza con un modelo. Un modelo es una descripción física o matemática de un sistema y usualmente representa un punto particular de acción en el tiempo (Beaverstock, Greenwood, Nordgren, & Nordgren, 2012). Asimismo, los modelos de simulación son empleados para estudiar y determinar la representación de un sistema real de manera abstracta con la intención de predecir el comportamiento de este. Sin embargo, dado que un sistema es un conjunto complejo, integrado de elementos interconectados, el cual forma parte de otro sistema superior y se compone de sistemas de orden anterior (Acosta Flores, 2007), se dificulta el nivel de abstracción pues la mayoría de los sistemas reales son muy complejos a para la realización de evaluaciones analíticas, por lo que los sistemas deben ser estudiados mediante la simulación (Law Averill, M & Associates Inc., 2007).

Un modelo de simulación representa la operación de algún proceso o sistema del mundo real, a través del tiempo, ya sea realizada manualmente o por medio de una computadora. La simulación involucra la generación de una historia artificial de un sistema y su observación para obtener inferencias relacionadas con las características operativas del sistema real (Banks, Carson, Nicol, & Nicol, 2005). Este modelo, comúnmente toma la forma de un conjunto de supuestos respecto a la operación del sistema, estos supuestos son expresados en forma de relaciones matemáticas y lógicas entre los objetos de interés del sistema. Entonces el sistema es modelado identificando las características de sus eventos y mostrando un juego de rutinas que dan una descripción detallada del sistema en estudio. Es así, como los eventos de la simulación evolucionan en el tiempo (reloj de simulación) ejecutando los eventos en orden creciente del tiempo de ocurrencia: un evento es algo que sucede en un instante de tiempo (simulado) que puede cambiar atributos, variables o acumuladores estadísticos. Asimismo, un evento discreto es un momento específico del tiempo (Winston, 2005). De esta forma, el estado de los

elementos de un sistema cambia continuamente o sólo en determinados estados del tiempo. El flujo del agua que entra y sale en una planta tratadora de aguas residuales y, la entrada y salida de clientes que pagan la despensa en un cajero son ejemplos de cambios continuos y discretos, respectivamente. Esto es, en el primer ejemplo el agua no muestra un espacio de tiempo entre litro y litro, por otro lado, las llegadas de las personas al cajero de un supermercado muestran tiempos variados entre una y otra.

Posteriormente, una vez desarrollado, validado y verificado el modelo de simulación, éste puede ser utilizado para investigar una amplia variedad de preguntas del tipo ¿Qué pasa si...?, acerca del mundo real (Kelton, Sadowski, & Sturrock, 2008). Una vez realizado lo anterior, se procede con la planeación de los experimentos que se harán mediante el modelo de simulación establecido. Después de haber concluido el modelo de simulación, los clientes o dueños del sistema real, tendrán una herramienta que les permitirá proyectar distintos escenarios de simulación en diferentes periodos de tiempo. La simulación también puede ser utilizada para estudiar sistemas en su etapa de diseño (antes de que dichos sistemas sean construidos). De esta manera, los modelos de simulación pueden ser utilizados tanto como un instrumento de análisis para predecir los efectos de cambios en sistemas existentes en lapsos de tiempo distintos, o bien como un instrumento de diseño para predecir el comportamiento de sistemas nuevos en proyecciones que contribuyan a soluciones alternativas en efectos presentes para un futuro sustentable.

Se considera que un sistema es modelado y simulado con la finalidad de identificar sus componentes, su estructura y observar su conducta en términos de sinergia, con el propósito de crear, planificar, controlar y predecir el comportamiento de distintos escenarios.

La simulación en el tiempo

La palabra simulación, en su concepto actual, se remonta hacia fines de 1940 cuando Von Neumann y Ulam definieron el término "análisis de Monte Carlo" para aplicarlo a una técnica matemática que usaban al resolver ciertos problemas de protección nuclear que eran demasiado costosos o muy complicados para resolverse experimental o analíticamente (Aspray, 1993). Sin embargo, la simulación analógica fue muy evidente

durante los años 50, la cual se definía entonces como Simulation Programming Languages (SPL) o bien simulación por ordenador (Nance, 1993).

La necesidad de resultados cada vez más precisos y rápidos estimuló el desarrollo de cientos de nuevas técnicas de simulación en los últimos 60 años. Además, con el desarrollo de la computadora de gran velocidad, a principios de 1950, la simulación tomó otro significado, ya que surgió la posibilidad de experimentar con modelos matemáticos (que describen sistemas de interés) en una computadora. Su sorprendente evolución brindó la herramienta necesaria que permite la implementación de los métodos de simulación más complejos de una manera simple y eficiente (Beaverstock, Greenwood, Nordgren, & Nordgren, 2012). El primer simulador fue General Simulation Program (GSP) acreditado por K.D. Tocher en conjunto con D.G. Owen en 1960 (Tocher & Owen, 1960). Mientras que en el periodo de 1961 a 1965 surgieron nuevos simuladores los cuales incluían declaraciones del principal lenguaje de programación llamada FORTRAN (Formula Translator System es un lenguaje de programación adaptado especialmente al cálculo numérico y a la computación científica desarrollado por John Backus e IBM en 1954), el cual se tenía como idioma hasta entonces (Barceló G, 2008): General Purpose System Simulator (GPSS) y SIMULA descritos por Wexelblatt (Wexelblatt, 1978), SIMSCRIPT enunciado por Marcowitz et. al. (Marcowitz, Hausner, & Karr, 1963), Control and Simulation Language (CSL) el cual contribuía con la solución de problemas complejos dentro de las empresas industriales y comerciales (Esso, 1963) y algunos más enunciados en la publicación de Nance (Nance, 1993). En esa época el diseñador del software utilizaba instrucciones de lenguaje y macros para especificar al sistema su lógica de operación. Por lo que el modelador tenía inicialmente que entender todo el conjunto de argumentos antes de poder hacer un seguimiento de las variables de estado del sistema. Esto implicaba la construcción de macros complicados, se requería de mucho tiempo y los costos eran excesivos. Además, los modelos que empleaban estos lenguajes de simulación no contaban con animación (Zeigler, Gon Kim, & H, 2000). Posteriormente, la aparición de un nuevo lenguaje popular y de uso general llamado PASCAL desarrollado por Niklaus Wirth entre los años 1968 - 1969, estimuló el diseño posterior de paquetes

de simulación basados en este lenguaje (Nance, 1993): Bryant (Bryant, 1980) (Bryant R. M., 1981) desarrolló el lenguaje de programación SIMPAS como un lenguaje de simulación de redes el cual fue diseñado para ser altamente portátil y orientado a eventos. Otro programa sobresaliente fue el lenguaje de simulación de redes llamando INTERACTIVE, el cual permitía la utilización de símbolos gráficos además de que la construcción y la ejecución eran interactivos (Lakshmanan, 1893).

Por otro lado, un primer lenguaje que permitió añadir animación fue SIMAN el cual fue creado por C. Dennis Pegden en 1982. Sin embargo, para crear animación SIMAN utilizaba un segundo lenguaje independiente llamado CINEMA (Marcowitz, Hausner, & Karr, 1963). SIMAN y CINEMA más tarde (en el año 2000) se combinaron para crear al software

Arena.

En 1986, WITNESS lanzó el primer producto con nivel de construcciones prefabricadas junto con una mejor representación visual. Este nuevo software redujo aún más la necesidad de una programación directa en el lenguaje de simulación. La nueva generación de simuladores hasta entonces era más fácil de usar, pero aún carecían de flexibilidad y la programación de código en el ordenador no estaba disponible para el usuario. A finales de los años 80 existían en el mercado diferentes simuladores para computadora basados en Disk Operating System (DOS) los cuales fueron creados principalmente para usos comerciales, incluyendo WITNESS, ProModel, SLAM y SIMFACTORY. En esa misma época, existieron en el mercado una amplia variedad de software de simulación; no obstante, muchos de ellos eran difíciles de usar, debido limitada capacidad gráfica, lo que ocasionaba el uso máximo de la capacidad de la computadora al momento de ejecutar los modelos de simulación (Beaverstock, Greenwood, Nordgren, & Nordgren, 2012).

La nueva generación de software para simulaciones

En la década de los 90, surgieron diferentes simuladores. En Europa surgió el Simple++, Simul8 y Taylor II (el primer simulador 3D para ejecutarse en una PC). En Estado Unidos, se crearon muchos programas de simulación de los cuales los más notables son: Extend y Simcad. En 1998 Taylor ED fue puesto en el mercado. Este último software se destacó

por ser el primer software de simulación orientado a objetos en 3D con realidad virtual, el cual también operaba con la plataforma del software de Microsoft Windows. Posteriormente, en el año 2003 fue lanzado el software FlexSim, el cual resultó ser sustancialmente diferente respecto a los simuladores anteriores tanto en su lenguaje de simulación como en su arquitectura (Beaverstock, Greenwood, Nordgren, & Nordgren, 2012).

MÉTODO

Simulación con Flexsim

Descripción del software. El software FlexSim fue desarrollado por Bill Nordgren, Cliff King, Roger Hullinger, Eamonn Lavery y Anthony Johnson. FlexSim permite modelar y entender con precisión los problemas básicos de un sistema sin la necesidad de programaciones complicadas, esto debido a que ofrece una forma sencilla al desarrollar el modelo de simulación. Se enlistan algunas razones por las cuales FlexSim es una buena alternativa como herramienta en simulación:

Su amplia sección de preconstruidos permite abordar situaciones mucho más complejas sin tener que escribir código de software.

El software se orienta a objetos lo que admite una mayor visualización del flujo de producción.

Todo el proyecto se desarrolla en un ambiente tridimensional (3D), además de permitir importar infinidad de objetos de distintos paquetes de diseño, incluyendo AutoCAD, Solid Works, Catia, 3D Studio, Revit, Google Sketch-Up, etc.

Otra razón importante es que no sólo se pueden simular sistemas discretos, sino que también se admite la simulación de fluidos o modelos combinados continuo-discreto.

La generación de distintos escenarios y condiciones variadas son fáciles de programar.

Las distribuciones de probabilidad se pueden representar con gran precisión en lugar de valores promedio para representar fielmente la realidad.

Las gráficas, los reportes y todo lo que se refiere a los estadísticos se puede revisar a detalle.

Aplicaciones de FlexSim

FlexSim ha contribuido con aplicaciones de clase mundial en temas de medicina, salud, sistemas de logística tales como operaciones de contenedores en puertos, simulaciones distribuidas en varios equipos dentro de una empresa manufacturera, en la minería, en centros aeroespaciales e incluso se ha adaptado a la industria del servicio (hoteles, hospitales, supermercados, o muchas otras industrias) para simular la administración y operación de los recursos humanos.

Así mismo se ha demostrado en diferentes casos de diversos corporativos alrededor del mundo (Volkswagen, FEMSA, DHL, MICHELIN, MATTEL, VALE, CATERPILLAR, DISCOVER, ARMY, U.S. AIR FORCE, DUPONT, ABB, CARRIER, ORACLE, Tetra Pak, IBM, NASA, FedEx, AVON, Whirlpool, ALCAN, Remington, BAKER HUGHES, etc.) que FlexSim es una herramienta clave para mejorar los resultados al dar respuesta acertadas a los problemas planteados (FlexSim Software Products, Inc, 2012).

Uso y manejo del software Flexsim

En la presente sección, para poder describir el uso y manejo del software se desarrollará un caso sencillo de estudio de un modelo de múltiples estaciones de prueba.

En muchos modelos se ha ilustrado cómo manipular materiales que se encuentran en cantidades discretas (piezas, esferas, personas, etc.), pero dentro de muchos sistemas reales los materiales no necesariamente son piezas discretas. En este apartado se presentan los conceptos fundamentales para construir un modelo básico de estaciones de prueba. La construcción de un modelo con estas características requiere de mucha atención ya que existen detalles que no pueden pasarse por alto.

Descripción del caso de estudio

Una planta industrial tiene un sistema de producción con un único Queue FIFO (First-In-First-Out) alimenta cuatro estaciones de prueba en paralelo. El producto llega a la Queue cada 21 segundos desde el source. El tiempo de ciclo de pruebas es de 1 minuto. Las estaciones de prueba paran cada 20 minutos exponencialmente distribuidos y lleva entre 2 y 5 minutos uniformemente distribuidos arreglarlos. La tasa de fallos en las estaciones es del 10%. Las piezas que fallan se separan manualmente en una mesa de retrabajo,

con un tiempo lognormal (35.4, 3.2, 0.1) y se vuelven a introducir nuevamente al Queue que alimenta a las estaciones de prueba.

Descripción de los objetos del modelo

El Source presenta un tiempo de arribo de 21 segundo y un flujo de primeras disponibles hacia el Queue 2 con un máximo de capacidad de contenido de 50 piezas.

El Queue 2 presenta un flujo de producción de tipo Matching Itemtypes donde existe una separación de productos para ser asignados a los Queue 4, 5, 6, 7 respectivamente.

Cada una de las Queues tiene una capacidad máxima de 50 productos dependiendo su tipo y presenta un flujo de proceso de tipo First Available y luego el producto es enviado hacia el área de pruebas.

En los procesos se presentan un 10% de fallos y tiempo de proceso de 60 segundos. Se tiene un supervisor asignado para separar las piezas con defectos y volver a introducir nuevamente al queue que alimenta a las estaciones de prueba.

Los productos que cumplan con las pruebas serán enviados al Sink donde es la etapa final y termino del proceso.

En caso de no ser enviados al Sink por cumplimiento de las pruebas son enviados a una mesa de retrabajo Processor 14 para su análisis y al no cumplir con las especificaciones son enviadas al Queue 2 para empezar nuevamente el proceso de pruebas.

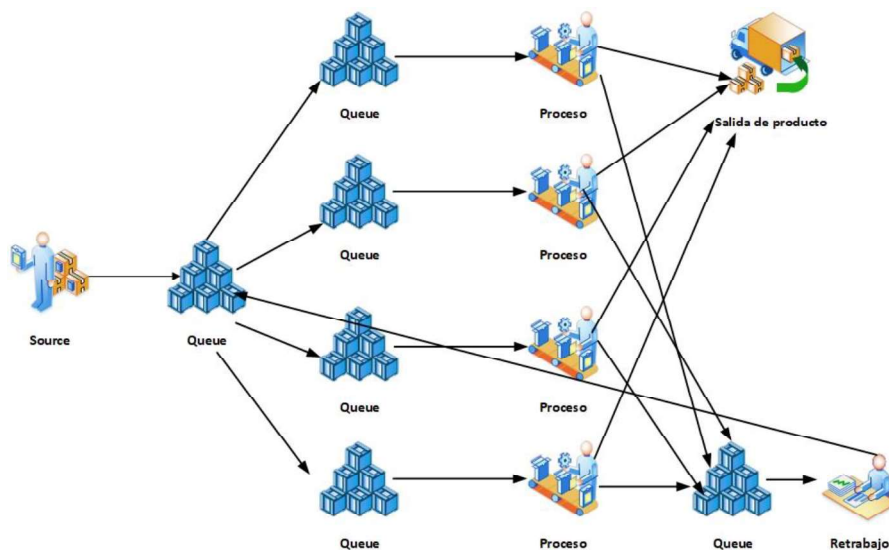


Fig. 1. Diagrama de flujo de producción del modelo de pruebas

Construcción del modelo

Para la construcción del modelo se inicia creando una nueva hoja de trabajo (New Model) en Flexsim. Posteriormente, se seleccionan de las librerías los recursos necesarios y se arrastran al área de trabajo (figura 2). Los objetos necesarios para este caso de estudio son:

- 1 Source que es la fuente de salida de producto hacia la línea de espera Queue 2
- 6 Queues que son las líneas de espera que son los envíos hacia las operaciones o procesos
- 5 procesos que son los encargados de las pruebas respectivas hacia los productos
- 1 Sink que es la salida de producto terminado y que cumple con las correctamente con las especificaciones.
- 1 Transporter (supervisor)

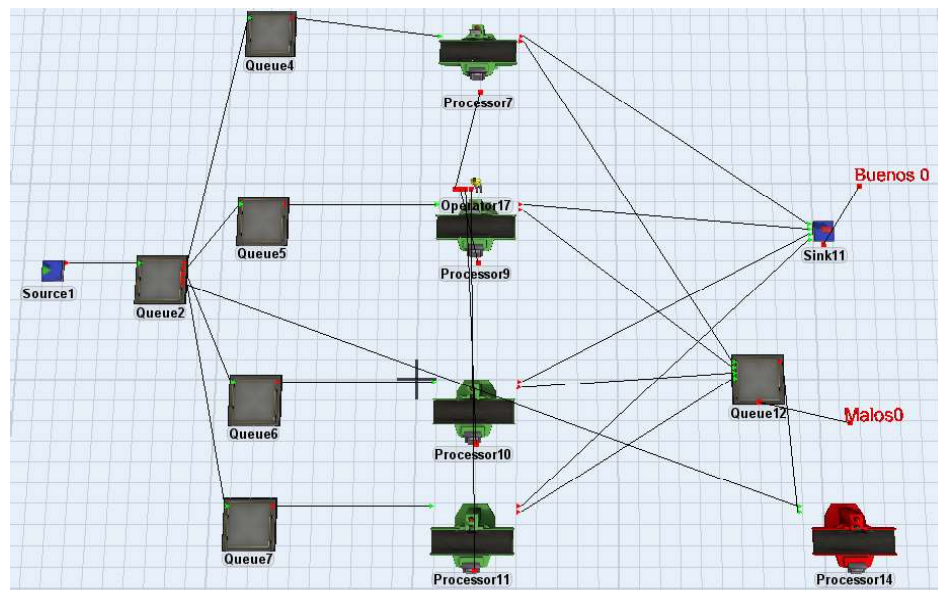


Fig. 2. Imagen del modelo final en ejecución

Conexión de objetos

Después de arrastrar los objetos al área de trabajo, es necesario conectarlos. La conexión de objetos fluidos se hace de la misma forma con la que conecta objetos discretos: presionando la tecla A y dando clic en los objetos a conectar se crea una conexión de entrada/salida y la tecla S crea una conexión de puerto central. Recuerde que los objetos

deben ser conectados de acuerdo con el diagrama de flujo de producción mostrado en la figura 1. Para este caso de estudio, las conexiones son las que se enlistan a continuación:

Tabla 1- Conexiones de elementos del sistema de modelación de pruebas

Actividad	Dependencia
Source 1	Queue 2
Queue 2	Queue 4
Queue 2	Queue 5
Queue 2	Queue 6
Queue 2	Queue 7
Queue 3	Processor 7
Queue 3	Processor 8
Queue 3	Processor 9
Queue 3	Processor 10
Processor 7	Sink 13
Processor 8	Sink 13
Processor 9	Sink 13
Processor 10	Sink 13
Processor 7	Queue 11
Processor 8	Queue 11
Processor 9	Queue 11
Processor 10	Queue 11
Queue 11	Queue 12

Configuración de los objetos

Configuración del Sources- Dado que el valor predeterminado del tiempo de llegadas es de 21 y el envío hacia el puerto Queue 2 es de First available.

Configuración de las líneas de espera (Queues). Habiendo posicionado los elementos de las líneas de espera se empieza la configuración dando doble clic en cualquiera de los elementos y se asigna primeramente la capacidad máxima que tendrá cada línea de espera y en este caso será de 50 cajas máximo y un envío (Send To Port) hacia las demás líneas de espera con la opción de Matching Itemtypes ya que en el caso de la Queue 2 es la primera línea de espera de servicio de los diferentes tipos de producto a enviar a las demás líneas de espera.

Configuración de los procesadores(Processors). Habiendo enviado los productos por las diferentes líneas de espera es ahora cuando deben ser sometidos a las pruebas de inspección para determinar si cumplen con las especificaciones y proceder a la salida de producto terminado. Se da clic en el procesador y en la opción de Processor se asigna una capacidad máxima de 1 caja por operación durante 10 segundos y se asigna un transporte quien realizara los ajustes manuales o cambios en este tiempo. En este caso de estudio se menciona que se presenta un 10% de defecto y esto significa que los envíos al reproceso son por probabilidad.

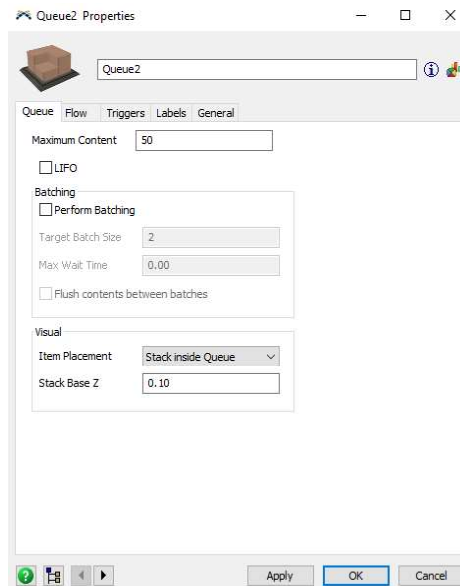


Fig. 3. Configuración de las propiedades de los Queue.

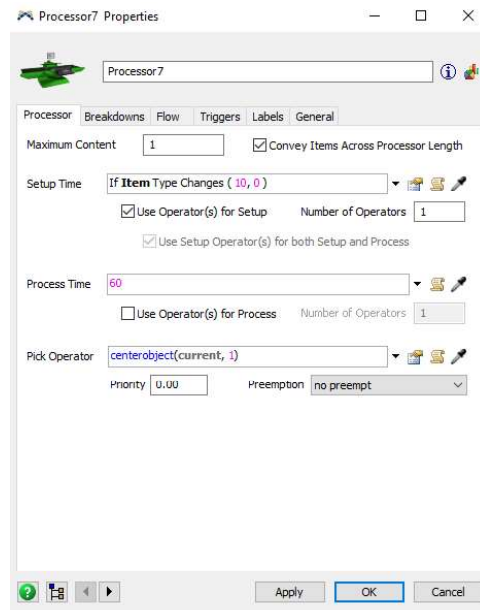


Fig. 4. Configuración de las propiedades de los Processors.

Configuración del Sink (Salida y recepción de producto final). Después de terminar las pruebas en los diferentes procesadores y las inspecciones del supervisor los productos son enviados al Sink donde este último toma como válido el producto terminado proveniente de los procesos y es almacenado. En caso contrario que después de realizar las pruebas pertinentes envíe el producto al Queue 12 esto significa que no cumple con las especificaciones y es enviado al Processor 14 que es la mesa de retrabajo que enviara el producto defectuoso a la línea de espera inicial del proceso de producción.

En el caso del Queue 12 y Sink 11 están programados mediante un elemento llamado VisualTool para llevar el conteo de los productos que cumplen (Buenos) y no cumplen (Malos) de acuerdo con la configuración del sistema y tener datos reales del comportamiento que tendría y así poder tomar las mejores decisiones en las operaciones.

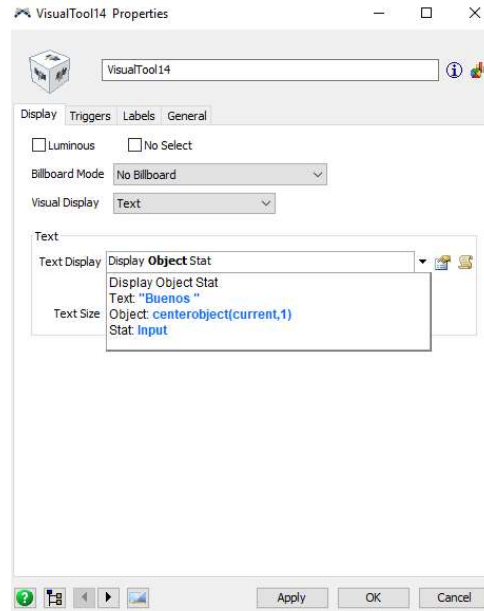


Fig. 5. Configuración de las propiedades del VisualTool.

Ejecución del modelo

Para ejecutar el modelo, definir el tiempo de simulación (por ejemplo 3600 segundos) y después dar Reset una vez que modelo esté corriendo, puede observarse como cada una de las líneas de espera cumple con su capacidad máxima de inicio y fin, así como los procesos en ejecución y el supervisor. También se puede observar la cantidad total de producto aceptado y rechazado durante el tiempo total de ejecución del sistema.

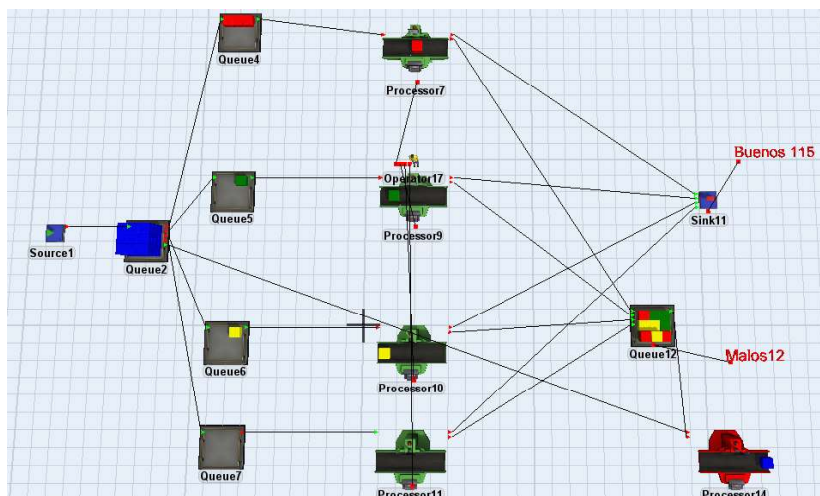


Fig. 6. Presentación del modelo final

RESULTADOS

El desempeño del sistema se describe con estadísticas. Las estadísticas son la acumulación de valores en una variable por un tiempo determinado, inician en 0 cada vez que se restaura (reset) el modelo y se acumulan valores hasta el momento en que se detiene la simulación. Además, estos resultados pueden ser vistos como números, porcentajes o gráficos y pueden programarse variables de interés para el usuario o simplemente consultarse las ya definidas.

Una vez que se ha ejecutado el modelo durante un período de tiempo predeterminado, se pueden analizar los resultados de la ejecución en muchas y muy variadas formas.

1. Algunas estadísticas que pueden ser observadas de forma rápida, se encuentran disponibles en los objetos que se muestran en el área de trabajo.
2. Además, también es posible agregar recursos especiales de la librería de objetos. Por ejemplo, el Recorder permite añadir gráficas dinámicas y variables, las cuales se animan mientras el modelo se está simulando.
3. En el menú Statistics > Reports and Statistics se puede tener acceso a un informe completo, a un resumen del informe o quizá a un informe del estado en el que se encuentra el modelo.

Finalmente, recordar que, para hacer una inferencia válida, los modelos de simulación se estabilizan haciendo ejecuciones con periodos de tiempo largos. Además, es importante tener en cuenta la importancia de ejecutar el número necesario de réplicas, puesto que las ejecuciones múltiples contribuyen también a mejorar la validez de la inferencia. Un mayor número de réplicas implica un número más elevado de muestras aleatorias independientes con distribución estadística diversas.

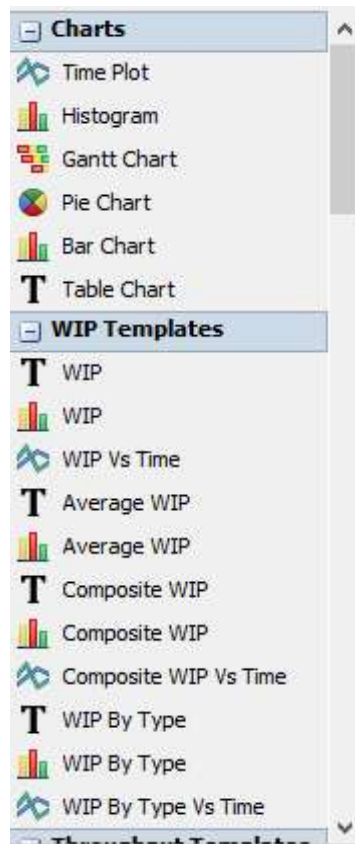


Fig. 7. Herramientas de informe de las estadísticas de operación de un modelo de simulación

CONCLUSIONES

Después de entender la terminología general de la simulación y de haber construido su propio modelo en FlexSim el modelador deberá tener idea clara de cómo operan los objetos en FlexSim. Además, como resultados de esta investigación, el lector comprobará que la simulación de un modelo desarrollado con FlexSim permite tomar mejores decisiones en la operación de los sistemas industriales, los cuales buscan mejorar su rentabilidad. Finalmente, se espera que este artículo refleje la importancia de la simulación tanto a investigadores, como a catedráticos y alumnos hacia el involucramiento en la construcción, análisis e interpretación de modelos más grandes, complejos y realistas, basados en la simulación con FlexSim.

LITERATURA CITADA

- Acosta Flores, J. (2007). *Ingeniería de Sistemas, un Enfoque Interdisciplinario*. México, D. F: AlfaOmega.
- Aspray, W. (1993). *Jhon Von Neumann y los orígenes de la Computación Moderna*. España: GEDISA.
- Banks, J. S., Carson, B. L., Nicol, N., & Nicol, D. M. (2005). *Discrete-Event System Simulation*. E.U: Prentice Hall.
- Barceló G, M. (2008). *International Conference on Operational Research. Una Historia de la Informática (págs. 50-68)*. barcelona, España: Rambla del Poblenou.
- Beaverstock, M., Greenwood, A. G., Nordgren, L., & Nordgren, W. (2012). *Applied Simulation Modeling and Analysis Using FlexSim*. Orem, E.U: FlexSim Software Products, Inc.
- Bryant, R. M. (1980). *A simulation Language Based on PASCAL*. *Proceeding of the 1980 Winter Simulation Conference (págs. 25-40)*. E. U: T. I. Oren, C. M. Shub and P. F. Roth.
- Bryant, R. M. (1981). *A Tutorial on Simulation programming with SIMPAS*, In *Proceeding of the 1981. Winter simulation Conference (págs. 363-377)*. E. U: T. I. Oren, C. M. Shub and P. F. Roth.
- Esso. (1963). *Reference Manual/Control and Simulation lenguaje*. E.U: Esso Petroleum Company, Ltd & IBM United Kingdom.
- FlexSim Software Products, Inc. (12 de Enero de 2012). *Improve Performance*. Obtenido de Saave money, Who is using Flexsim?: <http://www.flexsim.com>
- Kelton, D. W., Sadowski, R. P., & Sturrock, D. T. (2008). *Simulación con Software Arena*. México, D. F: Mc Graw Hill.
- Lakshmanan, R. (1893). *Desing and Implementation of a PASCAL Based Intaractive Network Simulation Language for Microcomputers*. Rochester, Michigan: Oakland, University.
- Law Averill, M & Associates Inc. (2007). *simulation Modeling and Analysis*. Arizona, E. U: Mc Graw Hill.
- Marcowitz, H. M., Hausner, B., & Karr, H. W. (1963). *SIMSCRIPT: A simulation Programming Language*, The RAND Corporation. Prentice Hall.
- Nance, R. E. (1993). *A History of Discrete Event Simulation Programming Languaje*. 4-39.
- Tocher, K. D., & Owen, D. G. (1960). *The Automatic Programming of Simulation*, *Proceeding of tha second*.

Wexelblatt, R. L. (1978). History of programming Language, Academic Press,. Proceedings of the ACM SIGPLAN History of Programming Languages, (págs. 1-3).

Winston, W. L. (2005). Investigación de Operaciones Aplicación y Algoritmos. México, D.F: Thompson Learning.

Zeigler, B., Gon Kim, T., & H, P. (2000). Theory of Modeling and Simulation. N. Y, E.U: Academic Press.