

# Simulación de Eventos Discretos



# Simulación de eventos discretos

Introducción . . . . .	3
Un ejemplo . . . . .	5
Procesos estocásticos . . . . .	8
Optimización de procesos . . . . .	11
DEVS en otros procesos . . . . .	14

## Introducción

Vamos a hablar ahora de la simulación de eventos discretos (DEVS, en inglés), una forma sencilla pero versátil de describir un sistema dinámico. Las simulaciones se basan en un modelo de un proceso del mundo real para imitar el comportamiento dependiente del tiempo. El enfoque de eventos discretos modela un proceso como una serie de sucesos instantáneos, o eventos discretos. Entre estos eventos, el sistema se aproxima como algo fijo e inmutable.

Podemos ilustrar este concepto desarrollando un modelo de ascensor. Un ascensor consta de un hueco, un mecanismo de elevación y una cabina, cada uno con su propio conjunto de subcomponentes. Tenemos que decidir cuáles de estos detalles vamos a incluir en nuestro modelo. ¿Es importante saber cómo afecta la pulsación de un botón en el panel de control de la cabina a cosas como la corriente en el circuito, la rotación del motor y el desplazamiento de la puerta? ¿Necesitamos entender estos detalles milisegundo a milisegundo? Tal vez. Esa información sería importante si se utilizara la simulación para diseñar el software del microcontrolador del ascensor. Pero si la tarea fuera diferente, si en lugar de eso quisiéramos determinar la capacidad del ascensor para acomodar el tráfico de pasajeros, entonces modelar el comportamiento a nivel de milisegundos probablemente no sea necesario.

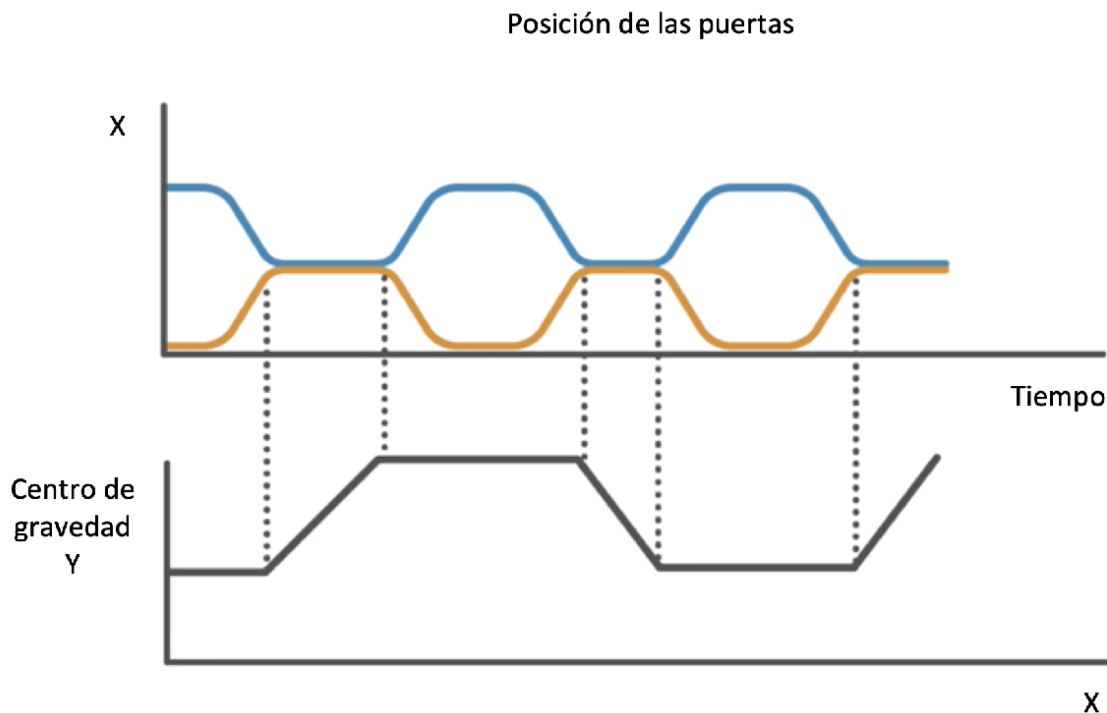
Por lo tanto, vamos a centrarnos en la serie de eventos que describen el comportamiento del ascensor. Empezaremos con:

- la apertura de la puerta.
- Una vez completada esa tarea, los pasajeros entran en el ascensor y pulsán un botón.
- A continuación, las puertas comienzan a cerrarse y el ascensor se pone en marcha.
- A continuación, la cabina se detiene en la planta deseada,

- las puertas se abren, los pasajeros salen y
- la puerta se vuelve a cerrar.

Entre estos eventos, las cosas cambian continuamente en el mundo real. Sin embargo, una simulación de eventos discretos ignoraría esta dinámica y consideraría que los pasajeros se encuentran en un estado fijo cuando no experimentan ningún evento. Todo lo que ocurra entre los eventos es intrascendente, lo que simplifica el modelo y nos permite centrarnos en la información que nos importa: cosas como el tiempo de tránsito, la utilización del ascensor y el rendimiento del movimiento de los pasajeros.

Podemos contrastar una simulación dinámica continua con una simulación de eventos discretos trazando un conjunto hipotético de resultados para ambas. Aquí tenemos tres líneas que representan las posiciones de las puertas y el centro de gravedad de la cabina.



Cualquier valor en el eje Y es aceptable en una representación continua. Pero con un enfoque de eventos discretos, el eje Y estaría compuesto por estados discretos, y los cambios de estado corresponderían a eventos discretos. En este caso, todos los eventos ocurren en diferentes momentos, pero no hay nada que impida que se produzcan múltiples eventos en un mismo momento. La simulación es, en realidad, un programa de eventos que se ejecutan uno tras otro, y el tiempo es sólo un artefacto que se controla entre bastidores. Otra forma de pensar en esto es organizar todo en un calendario de eventos. Algunas fechas tendrán múltiples eventos, mientras que otras no tendrán ninguna actividad.

Así que aquí tenemos uno de los muchos métodos para expresar visualmente una simulación de eventos discretos de un ascensor: uno que se centra en la experiencia de

los pasajeros. Los pasajeros se modelan como unidades individuales, o entidades, que se mueven a través del sistema del ascensor. Los pasajeros que esperan el ascensor se modelan como entidades sentadas en colas, un componente del modelo que mantiene las entidades hasta que un evento permite su salida. En este caso, la llegada del ascensor indica que los pasajeros pueden pasar al siguiente paso abriendo una puerta en el modelo. Una vez en el ascensor, el tránsito se modela con lo que se conoce como servidor.

El servidor retiene las entidades durante un tiempo determinado, que en nuestro caso es el tiempo que se tarda en llegar a la planta deseada.

Así, con conceptos básicos como:

- entidades,
- colas,
- puertas y
- servidores,

las simulaciones de eventos discretos nos permiten explorar cuestiones fundamentales sobre un proceso. En nuestro caso, podemos utilizar la simulación para responder a cuánto tiempo pasan los pasajeros en el ascensor, con qué frecuencia el ascensor está inactivo y qué plantas demandan más uso. Pero las cuestiones relativas a:

- la latencia del sistema,
- la utilización de los recursos y
- la identificación de los cuellos de botella

son tan omnipresentes que la simulación de eventos discretos se aplica a problemas tan diversos como los servicios de los cajeros de los bancos o el tráfico de las redes de comunicación. Podemos profundizar en estos temas en otra ocasión, pero por ahora, ya tenemos la visión general.

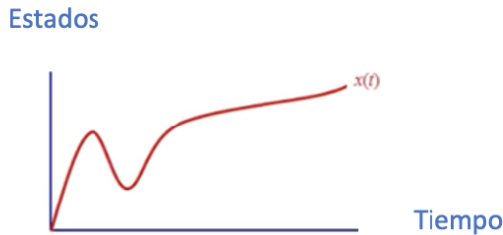
## Un ejemplo

La simulación de eventos discretos analiza el comportamiento de un sistema dinámico aproximándolo como una secuencia de sucesos instantáneos. Examinemos por qué son tan potentes para ciertas aplicaciones y por qué podría utilizarlas en lugar de otras técnicas de simulación. Algunos procesos se prestan bien a la simulación de eventos discretos debido a su naturaleza basada en eventos. En situaciones en las que la elección es menos clara, puede adoptar un enfoque de eventos discretos debido a las ventajas computacionales que ofrece sobre una simulación de dinámica continua. En última instancia, sin embargo, la adopción dependerá del problema que se intente resolver. Las simulaciones de eventos discretos se utilizan a menudo para responder a cuestiones relacionadas con:

- la programación,

- la asignación de recursos y
- la planificación de la capacidad.

## Continuo



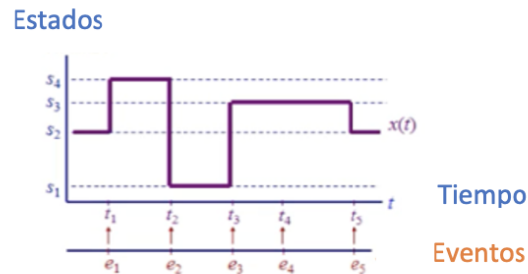
ESPACIO DE ESTADOS:

$$X = \mathbb{R}$$

DINÁMICA:

$$\dot{x} = f(x, t)$$

## Discreto



ESPACIO DE ESTADOS:

$$X = \{s_1, s_2, s_3, s_4\}$$

DINÁMICA:

$$x' = f(x, e)$$

El estadístico George E. P. Box escribió que «todos los modelos son erróneos, pero algunos son útiles». Entender el propósito de su simulación dicta cómo aproximar el sistema. En muchos casos, la simulación de eventos discretos es una forma directa de modelar el problema y adquirir los datos deseados. Por ejemplo, si su tarea es entender cómo el número de cajeros afecta a la longitud de las colas en una tienda de comestibles, probablemente no se preocuparía de simular cada pasada de código de barras. Si quiere predecir las tasas de caída de bits en una red de datos, probablemente no se preocupe por el voltaje de cada transistor. Es en este tipo de aplicaciones, cuando entran en juego cuestiones como la gestión de recursos, cuando los detalles de bajo nivel se vuelven irrelevantes y las simulaciones de eventos discretos resultan útiles. Veámoslo más de cerca.

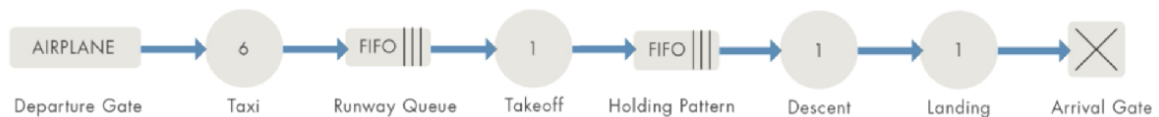
Consideremos la tarea de simular el vuelo de un avión. Un enfoque sería una simulación basada en la trayectoria en la que el modelo sigue diligentemente la posición del avión. Se podría ejecutar esta simulación en cualquier punto del tiempo y conocer con precisión la historia de la latitud, la longitud y la altitud del vehículo a través de todas las etapas del vuelo.

Pero tal vez no nos preocupe sólo un avión. Tal vez queramos simular toda una flota de aviones que vuelan entre decenas de aeropuertos en varios países. Y el motivo de la simulación es que queremos entender cómo las condiciones meteorológicas y los retrasos

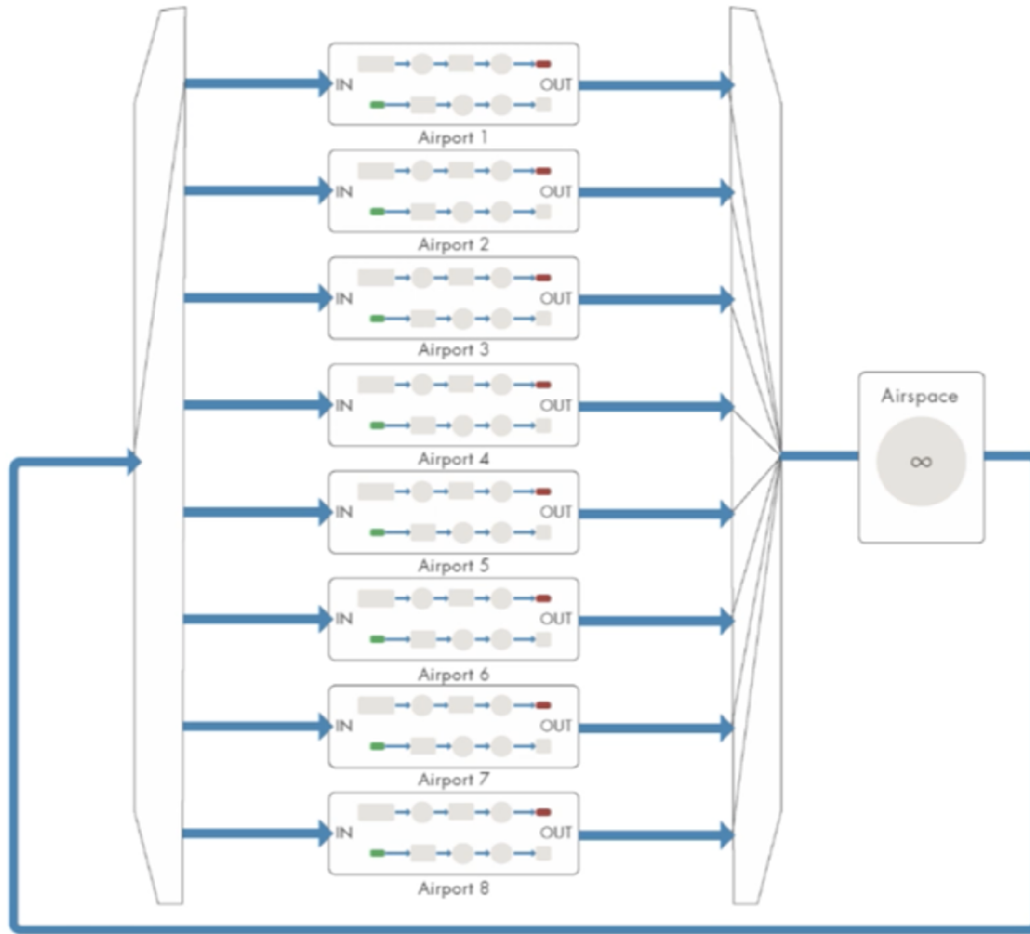
del tráfico aéreo en una región afectan al rendimiento de la flota en todo el mundo. Si este fuera el objetivo, la única información relevante es qué hacen los aviones y en qué región se encuentran. Calcular la latitud, la longitud y la altitud de cada avión en cada momento es un despilfarro computacional. Simplemente no necesitamos todos esos datos para obtener la respuesta que buscamos.

Así que, en su lugar, vamos a registrar únicamente las fases de vuelo en las que se encuentran las aeronaves. Este tipo de modelización nos permite utilizar una simulación de eventos discretos, en la que los eventos son las transiciones entre las fases de vuelo.

El tiempo transcurrido en cada fase se representa mediante servidores, y las colas representan los retrasos en tierra y los patrones de espera. Al simplificar las cosas, el esfuerzo de un ingeniero de diseño para desarrollar el modelo puede reducirse considerablemente.



Además, la carga computacional de la simulación se reduce drásticamente, ya que los únicos cálculos realizados son las actualizaciones de la fase de vuelo de cada avión. Esto significa que podemos realizar más simulaciones, lo que nos proporciona una imagen más completa del sistema en diferentes escenarios.

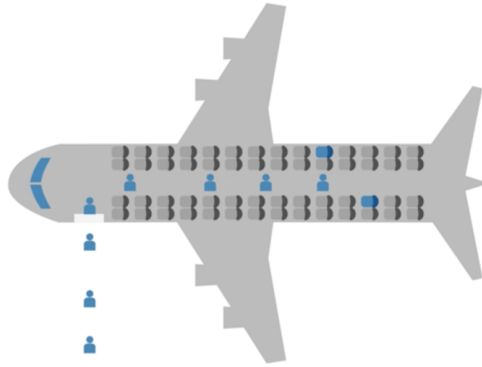


## Procesos estocásticos

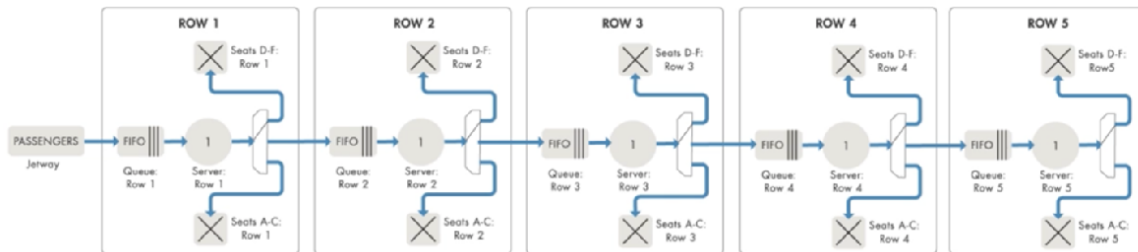
Vamos a hablar de los procesos estocásticos en el contexto de la simulación de eventos discretos. Un proceso estocástico es aquel en el que los aspectos del sistema son aleatorios. Dado que no hay un resultado definitivo sobre cómo evolucionarán los procesos a lo largo del tiempo, a menudo se denominan «no deterministas». Los procesos estocásticos son especialmente importantes para la simulación de eventos discretos como método de aproximación a los detalles de un sistema que no podemos o decidimos no modelar. Si no tenemos en cuenta estos detalles y definimos todos los parámetros de nuestro modelo como constantes, la simulación sería trivial y poco informativa.

Para ilustrar este concepto, consideremos una simulación de eventos discretos de pasajeros que suben a un avión.



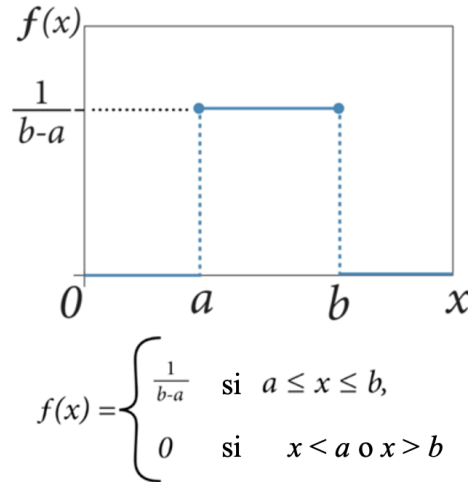


Una forma de conseguirlo es modelar el pasillo como una serie de colas y servidores por los que las entidades, en este caso los pasajeros, se mueven hasta llegar a su asiento asignado. Cuando llegan a la fila correcta, los pasajeros guardan su equipaje de mano en el compartimento superior antes de dirigirse a su asiento. Lo único que hay que hacer es definir el tiempo necesario para que cada pasajero realice estas tareas para simular el tiempo que tarda el avión en embarcar completamente.



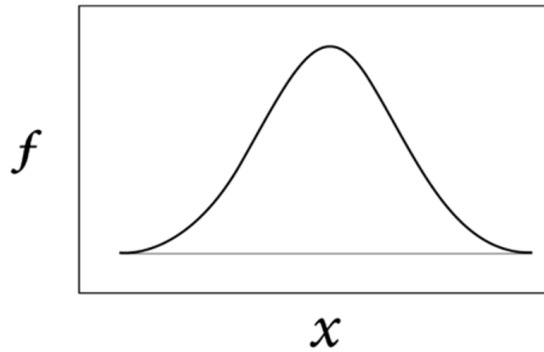
Una aproximación de primer orden a este proceso consiste en suponer que cada pasajero tarda exactamente el mismo tiempo en completar la tarea de guardar el equipaje de mano y colocarse en su asiento. Pero todos sabemos por experiencia personal que esto no es así; algunas personas son más lentas que otras. Por lo tanto, una simulación debe modelar la variabilidad en la duración de las tareas para proporcionar resultados más significativos. La cuestión es cómo hacerlo de la mejor manera posible. No es posible modelar todos los matices del comportamiento de una persona cuando se sitúa en su asiento. Pero podemos acercarnos más a la realidad si aleatorizamos el tiempo que cada pasajero pasa en los servidores.

Ahora, por supuesto, tenemos que poner algunas restricciones en el modelo para que los valores aleatorios sean razonables. Podemos conseguirlo definiendo una distribución de probabilidad  $X_i = P(X = x) = f(x)$  para el tiempo que una entidad pasa en un servidor. La distribución no es más que las probabilidades de que se seleccione un número concreto. Una estrategia sería utilizar una distribución uniforme:



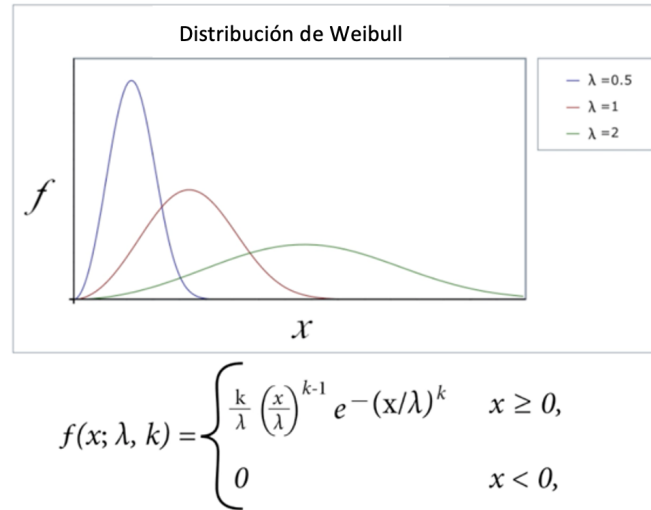
en la que se colocan las mismas probabilidades en cada valor dentro de un rango especificado. En nuestro caso, podríamos decir que los pasajeros tardan entre 2 y 10 segundos en sentarse.

Pero si se midiera realmente el tiempo que los pasajeros tardan en realizar esta tarea, se podría comprobar que un mayor número de ellos se agrupa en torno a un valor concreto en el centro de este intervalo, mientras que se encuentran menos en los extremos. Este es un resultado estadístico común, por lo que a menudo se ven distribuciones gaussianas o normales en los modelos.



$$f(x \mid \mu, \sigma^2) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Sin embargo, en el caso de la carga de pasajeros, una distribución gaussiana probablemente no sea la mejor opción. Dado que es imposible que una tarea dure menos de cero segundos, una distribución Poisson o Weibull podría tener más sentido. Pero la distribución por la que se opte dependerá del fenómeno que se intente caracterizar.



Ahora bien, no es todo o nada cuando se trata de emplear la probabilidad en simulaciones de eventos discretos. Hay que elegir cuánto modelar de forma determinista y confiar en la probabilidad para rellenar el resto. En general, hay que centrarse en incluir los detalles del sistema que no se ajustan bien a una distribución de probabilidad. Por ejemplo, el tiempo necesario para acceder a un asiento en un avión depende en gran medida de si un pasajero sentado está o no en el camino. Si esa persona tiene que levantarse para hacer sitio, la duración del proceso de sentarse aumenta considerablemente. Así que, en este caso, lo que realmente se quiere es emplear probabilidades específicas para la situación en lugar de una regla única y general.

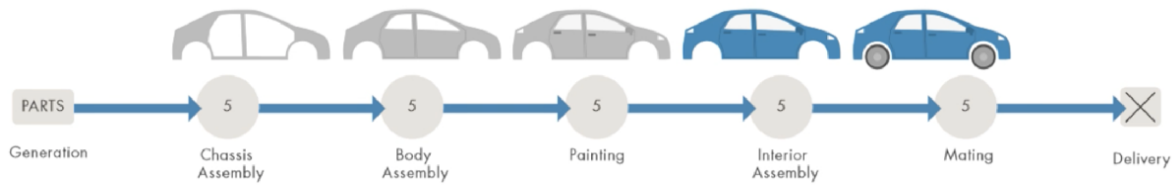
Esta técnica de mezclar determinismo y no determinismo es la que hace que las simulaciones de eventos discretos sean tan valiosas. La colocación juiciosa de los términos probabilísticos permite realizar análisis significativos sin complicar demasiado el modelo.

## Optimización de procesos

Hablemos ahora de la simulación de eventos discretos en el contexto de la optimización de procesos. La optimización de procesos es un tema amplio que implica el examen de los procesos creados por el hombre con el objetivo de mejorar su rendimiento. Todo proceso requiere recursos como tiempo, dinero, materiales, equipos y personal. La optimización se lleva a cabo para poder tomar decisiones inteligentes sobre la mejor manera de utilizar estos recursos con el fin de satisfacer sus objetivos - cosas como maximizar la productividad y los ingresos - al tiempo que se minimizan los defectos y los costes. Y, por supuesto, hay muchas maneras de analizar una operación, pero como muchos procesos pueden abstraerse fácilmente como sistemas impulsados por eventos, a menudo se utiliza la simulación de eventos discretos. Veámoslo más de cerca.

Una de las numerosas disciplinas de la optimización de procesos es la fabricación. Una cadena de montaje es un buen candidato para una simulación de eventos discretos porque puede dividirse en una serie de pasos finitos. Si nuestra cadena de montaje de

coches incluyera cinco pasos, uno tras otro,



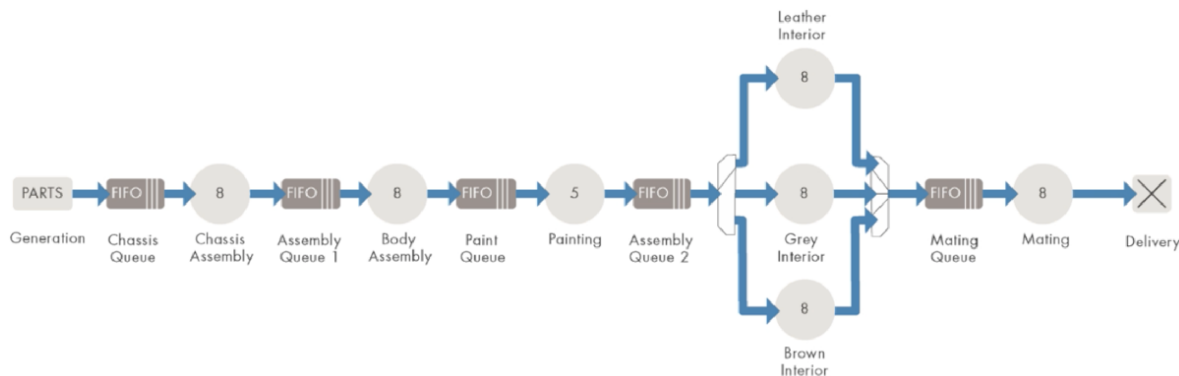
podríamos representar los coches como entidades que pasan por cinco bloques de servidores en serie. Dado que nuestro objetivo es comprender los objetivos de alto nivel, como la tasa de producción, los detalles de cada uno de estos pasos no son particularmente relevantes para nosotros. No importa cómo se pinta el coche. Sólo necesitamos modelar el tiempo que lleva ese y todos los demás pasos del proceso.

Ahora bien, cuando un coche se ensambla, todo está bien si puede pasar inmediatamente al siguiente paso. Pero si hay un retraso en un proceso de fabricación, eso puede ser modelado con colas.



Por supuesto, las colas suponen un desperdicio de tiempo de inactividad, por lo que una tarea habitual de la optimización de procesos es realizar un análisis coste-beneficio para aumentar el rendimiento de un paso concreto.

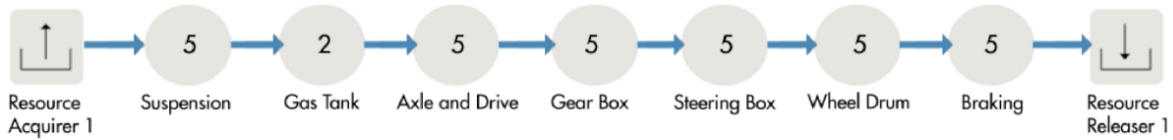
Tal vez podríamos disminuir el tiempo de montaje interior con mejor maquinaria, lo que se modela ajustando el tiempo de servicio. O, por el contrario, podríamos comprar más máquinas, lo que podría modelarse como un servidor con mayor capacidad de entidad o como servidores separados en paralelo.



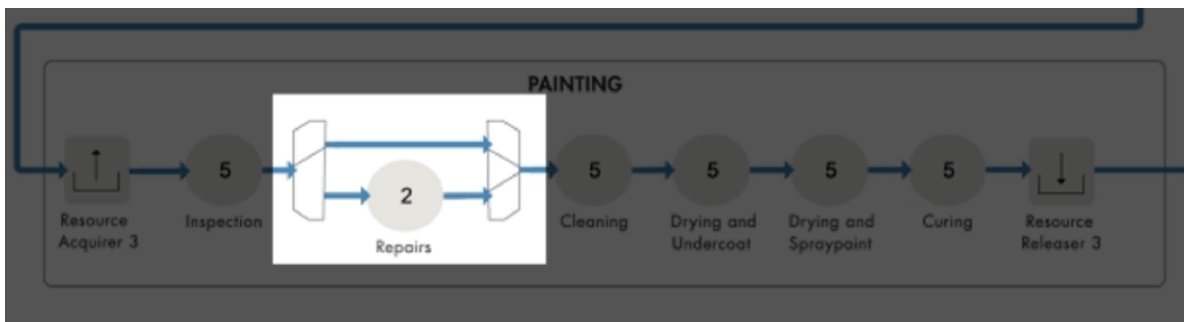
Por cierto, este último patrón de modelado es también la forma en que se modelaría el enrutamiento si diferentes automóviles tomaran diferentes caminos de fabricación.

Pero incluso con este matiz, este es el modelo más simple que se puede obtener para una fábrica de automóviles. La fidelidad del modelo podría aumentarse de varias maneras, y cada detalle mejoraría nuestra comprensión del sistema y nos permitiría tomar mejores

decisiones. Por ejemplo, podríamos desglosar las tareas en todas las subfases que las componen. También podríamos tener en cuenta la volatilidad de los plazos de estas tareas incluyendo términos probabilísticos en determinados pasos. Si se necesita personal u otros recursos en un punto concreto, podrían modelarse como un componente que se fusiona con el automóvil antes de que el proceso continúe.

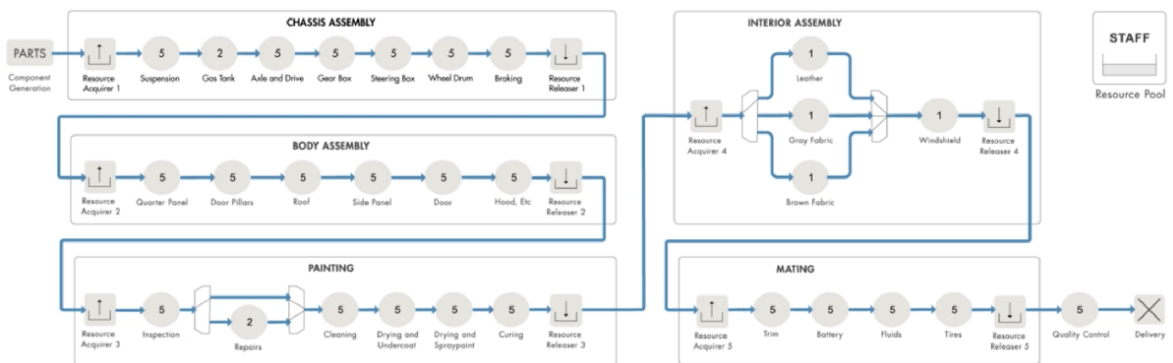


Se podrían insertar fallos y evaluar su impacto pausando o retrasando una acción en el modelo:



Y si la cadena de montaje se adapta a una situación cambiante, se puede alterar el rumbo en un intento de mejorar el rendimiento sobre la marcha, una simulación de eventos discretos tendrá que incluir un modelo de esa inteligencia. Quizá se modele la lógica de adaptación mediante un algoritmo escrito en código o como una máquina de estados finitos.

Cuando se incluyen todos estos detalles en una simulación de eventos discretos, se puede empezar a realizar potentes análisis de un sistema sobre el que, de otro modo, no se tendría ninguna intuición.



Una vez establecido el marco, se pueden ejecutar miles de escenarios diferentes y examinar cómo varía el resultado en función de aspectos como el programa de producción, los manifiestos de trabajo, la asignación de personal... lo que se quiera. Los resultados

de la simulación le permiten tomar decisiones informadas sobre la mejor manera de mejorar la operación con respecto a sus objetivos de rendimiento. Y cuando se vincula a un esquema de optimización numérica, el ordenador puede ayudarle a converger hacia el mejor resultado posible. Estas técnicas tienen un valor incalculable no sólo para la fabricación, sino para cualquiera de los ámbitos de la optimización de procesos.

## DEVS en otros procesos

Ahora examinaremos cómo puede utilizarse la simulación de eventos discretos para evaluar el rendimiento de los sistemas de comunicación digital. Con la electrónica cada vez más interconectada, los ingenieros se preocupan cada vez más por la transferencia de datos a través de las redes. Y tanto si se trata de un proveedor de servicios de cable como de un diseñador de aviónica, se quiere responder a preguntas similares... cosas como: «¿Tengo el ancho de banda necesario para soportar el tráfico de mi red?», «¿Es mi protocolo de software el mejor para la arquitectura de mi sistema?» o «¿Qué pasará con el rendimiento de las comunicaciones si añado, quito o sustituyo componentes en mi sistema?».

La simulación de eventos discretos es una buena manera de responder a estas preguntas sin tener que construir y probar una red real, porque los elementos estándar de un modelo pueden ser fácilmente asignados a un sistema de comunicación. Se pueden modelar paquetes de datos como entidades creadas por un dispositivo, transmitidas a través del medio de comunicación y recibidas por un dispositivo en el otro extremo. El canal de transmisión de datos puede modelarse como un servidor en el que el paquete reside durante un tiempo determinado que depende del tamaño del paquete, la distancia entre los dispositivos y el protocolo de comunicación.

Ahora bien, si se está modelando una comunicación bidireccional en la que las partes pueden tanto enviar como recibir información, entonces sería necesaria una segunda vía. Este esquema captura el comportamiento de la comunicación full duplex, lo que significa que las partes pueden enviar y recibir paquetes de datos simultáneamente. En esta configuración, no hay interferencias entre las señales, algo así como una calle de doble sentido.

Las limitaciones de ancho de banda suelen ser mayores en los sistemas de comunicación semidúplex, lo que significa que la comunicación sólo puede ir en una dirección en un momento dado. Este esquema es más parecido a un puente de un carril que soporta el tráfico en ambas direcciones. Y al igual que los coches pueden acumularse a cada lado del puente, también puede acumularse el tráfico en la red. Por ello, los sistemas de comunicación casi siempre incluyen búferes de datos que protegen de la colisión de paquetes entre sí, reteniendo la información hasta que sea seguro transmitirla. Este comportamiento puede modelarse con colas en la simulación de eventos discretos.

Pero el sistema tiene que identificar que el canal está en uso para que las partes sepan cuándo enviar y cuándo almacenar en el búfer. Este reto se conoce como Control de Acceso al Medio (MAC, en inglés), y la estrategia MAC de cada protocolo de co-

municación es un poco diferente. Independientemente de la lógica específica empleada, una simulación de eventos discretos capturará este tipo de enrutamiento de paquetes a través de una serie de bucles y callejones sin salida gestionados por conmutadores.

Elaborando aún más, podemos capturar el comportamiento de un primer sistema Ethernet. En él, cada estación está conectada a través de un cable compartido que admite la comunicación semidúplex. Las estaciones se basan en CSMA, Carrier Sense Multiple Access, para detectar la disponibilidad del canal. Si una estación quiere enviar datos y parece que el canal está libre, comienza la transmisión. Y mientras no se detecten colisiones de datos, se enviarán más bits hasta que se transmita todo el mensaje. Sin embargo, en caso de colisión, el protocolo Ethernet dicta que el emisor espere a que el canal se libere y lo intente de nuevo. El tiempo de espera o backoff es aleatorio para evitar que dos estaciones entren en conflicto una y otra vez.

Con suerte, la transmisión funcionará después de esto, pero si la estación sigue encontrando problemas, acabará rindiéndose y declarando un fallo. Y esto es exactamente el tipo de cosas que te gustaría descubrir como parte de una simulación de eventos discretos de una red de comunicaciones. En lugar de comprar todo el hardware y esperar lo mejor, hay que modelar la red para identificar los problemas con antelación. A continuación, se rediseña el modelo para mitigar los problemas, probando cosas como reducir el tráfico, alterar la estructura de la red o cambiar el protocolo de comunicación. Solucionar el problema en el modelo puede ser mucho más rápido y barato que hacerlo en el hardware. Y al iterar nuestro diseño en la simulación, podemos trabajar para optimizar el rendimiento frente a cualquier punto de referencia que elijamos.