

Vistas del Mundo (World Views) en la Simulación de Eventos Discretos (SED)

Juliana Quintana-Rojas

Departamento de Ingeniería Industrial
Universidad de Los Andes

World Views

Def. Las vistas del mundo son marcos conceptuales o enfoques de diseño que guían al analista sobre cómo estructurar la lógica, las variables y el flujo de tiempo dentro de un modelo de simulación

**Programación de
Eventos**

(Events scheduling)

**Interacción de
procesos**

(Process Interaction)

**Escaneo de
Actividades**

(Activity Scanning)

Programación de Eventos (Event Scheduling)

Def. Este enfoque se concentra directamente en los eventos individuales y en el efecto instantáneo que produce cada uno de ellos sobre las variables de estado del sistema. Un evento ocurre en un instante específico y ningún tiempo de simulación transcurre durante su ejecución lógica

Conceptos necesarios

- **Lista de Eventos Futuros (FEL):** Es la estructura central que contiene avisos de eventos (event notices) ordenados cronológicamente.
- **Reloj de Simulación (Clock):** El tiempo simulación avanza "saltando" de forma variable de un tiempo de evento al siguiente en la FEL, omitiendo periodos de inactividad

Interacción de Procesos (Process Interaction)

Def. Este enfoque se centra en el ciclo de vida de una entidad activa (como un cliente o una pieza) a medida que fluye a través de un sistema. Un proceso es una secuencia de eventos, actividades y retrasos asociados a una sola entidad.

Conceptos necesarios

- **Entidades dinámicas:** Objetos que se crean, fluyen por el modelo y finalmente se destruyen (por ejemplo, clientes).
- **Recursos limitados:** Elementos físicos o lógicos con capacidad restringida que las entidades deben solicitar, retener y liberar (por ejemplo, un cajero o una máquina)

- **Espera condicional (Delay) vs. incondicional (Activity)**

Una entidad realiza una espera incondicional cuando retiene un recurso por un tiempo conocido (actividad de servicio). Realiza una espera condicional cuando se ve obligada a hacer fila porque el recurso que necesita está ocupado por otra entidad

Escaneo de Actividades (Activity Scanning)

Def. Se concentra en las actividades del modelo y en las condiciones lógicas que permiten que una actividad comience

Conceptos necesarios

- **Enfoque Heurístico (Reglas If-Then):** Las actividades solo se inician si se cumplen ciertas condiciones lógicas del sistema (por ejemplo, "hay entidades esperando en la cola AND el servidor está libre")
- **Avance por incrementos fijos (FITA):** Clásicamente, el tiempo avanza en intervalos de tamaño constante (Δt). En cada paso, se escanean todas las condiciones de las actividades para ver cuáles pueden comenzar.

Ej. Cajero - Programación de eventos

Una tienda tiene una sola caja. Los clientes llegan y, si la caja está libre, los atienden de inmediato; si no, hacen fila. Se conocen los tiempos entre llegadas y los tiempos de servicio de los primeros 3 clientes.

- 1. Identificar los dos tipos de evento del sistema (Llegada y Fin_Servicio) y completar esta FEL a mano**
- 2. Llegadas: C1 en $t=0$, C2 en $t=2$, C3 en $t=5$. Tiempos de servicio: C1=4 min, C2=2 min, C3=3 min.**

Ej. Cajero - Interacción de procesos

El mismo cliente, visto ahora como una entidad que fluye por el sistema

1. **Dibujar el ciclo de vida de UN cliente típico con bloques:**
2. **Llegadas: C1 en $t=0$, C2 en $t=2$, C3 en $t=5$. Tiempos de servicio: C1=4 min, C2=2 min, C3=3 min.**
3. **¿En qué bloque hay una espera condicional (depende de que el recurso esté libre) y en cuál una espera incondicional (tiempo conocido de antemano)?**

Ej. Cajero - Escaneo de actividades

La actividad "atender cliente" solo puede iniciar si se cumplen dos condiciones a la vez.

SI (cola > _____) AND (cajero = _____) ENTONCES iniciar atención

- 1. Con $\Delta t = 1$ minuto, recorrer 4 pasos de tiempo en esta tabla marcando si la condición es verdadera o falsa.**
- 2. Llegadas: C1 en $t=0$, C2 en $t=2$, C3 en $t=5$. Tiempos de servicio: C1=4 min, C2=2 min, C3=3 min.**

Ej. Simulador único servidor

Se desea simular el comportamiento de un centro de atención con un único servidor (como una caja de supermercado o una barbería). El sistema inicia en estado completamente vacío y libre (sin clientes y con el servidor libre) a un tiempo $t = 0$. El objetivo de la simulación es observar el proceso de forma manual paso a paso hasta que $n = 6$ clientes hayan completado su retraso en la cola (la simulación finaliza en el instante en que el sexto cliente inicia su servicio)

Tiempos de llegada (t_i) e intervalos entre arribos (A_i)

- Cliente 1 llega a $t_1 = 0.4 \rightarrow A_1 = 0.4$
- Cliente 2 llega a $t_2 = 1.6 \rightarrow A_2 = 1.2$
- Cliente 3 llega a $t_3 = 2.1 \rightarrow A_3 = 0.5$
- Cliente 4 llega a $t_4 = 3.8 \rightarrow A_4 = 1.7$
- Cliente 5 llega a $t_5 = 4.0 \rightarrow A_5 = 0.2$
- Cliente 6 llega a $t_6 = 5.6 \rightarrow A_6 = 1.6$
- Cliente 7 llega a $t_7 = 5.8 \rightarrow A_7 = 0.2$
- Cliente 8 llega a $t_8 = 7.2 \rightarrow A_8 = 1.4$

Tiempos de servicio requeridos (S_i) por cada cliente

- $S_1 = 2.035$
- $S_2 = 0.73637$
- $S_3 = 0.237$
- $S_4 = 1.133$
- $S_5 = 3.733$
- $S_6 = 0.6$

Ej. Simulador único servidor

Ej. Simulador único servidor -Python

Se desea simular el comportamiento de un centro de atención con un único servidor (como una caja de supermercado o una barbería). El sistema inicia en estado completamente vacío y libre (sin clientes y con el servidor libre) a un tiempo $t = 0$. El objetivo de la simulación es observar el proceso de forma manual paso a paso hasta que $n = 6$ clientes hayan completado su retraso en la cola (la simulación finaliza en el instante en que el sexto cliente inicia su servicio)

1. Resolver el notebook simulación cola
2. Resolver el notebook simulación cola - extensiones