

# **Simulación a Eventos Discretos**

Tema 7: Construcción del modelo computacional

## **Software para construcción de un modelo de SED**

Usaremos AnyLogic, desarrollado por The AnyLogic Company.

Permite construir el modelo en un entorno gráfico y personalizar algunas de sus partes programando en Java.

En particular, usaremos la versión Personal Learning Edition, que restringe algunas funcionalidades y la cantidad de entidades que se pueden generar en una ejecución.

Disponible aquí.

Los modelos se almacenan en archivos de extensión `.alp`.

## Aprendizaje de AnyLogic

Muchas funcionalidades, algunas complejas (como todo software de SED).

Existen varios niveles de *expertise* en el uso del software. En este curso usaremos las funcionalidades necesarias para los objetivos del laboratorio.

Recursos disponibles aquí:

- Ilya Grigoryev. *AnyLogic in Three Days*. Sixth edition. Secciones recomendadas: Modeling and simulation modeling (p.7), Discrete-event modeling with AnyLogic (p. 133).
- Andrei Borshchev, Ilya Grigoryev. *The Big Book of Simulation Modeling*. Capítulos recomendados: Modeling and simulation modeling (ch. 1), Discrete events and the Event model element (ch. 8), Randomness in AnyLogic models (ch. 15), Model time, date and calendar. Virtual and real time (ch. 16).
- Arash Mahdavi. *The Art of Process-Centric Modeling with AnyLogic*. Para algunos aspectos particulares y en general para lectura avanzada.

## **AnyLogic: creación del modelo**

*File/New/Model:* Crea una ventana *Main* donde se agregan los componentes. Se debe especificar la unidad de tiempo del modelo.

*File/New/Agent Type:* Puede tener una imagen que lo representa y parámetros.

## AnyLogic: construcción del modelo (en general)

El modelo se construye desde la pestaña *Palette*. Componentes relevantes (columna izquierda de íconos, pestaña contextual *Properties* a la derecha):

- *Presentation*: Layout estático, imágenes de fondo.
- *Process Modelling Library/Space Markup*: Aspecto estático del modelo. Discretización del espacio, redes de desplazamiento.
- *Process Modelling Library/Blocks*: Aspecto dinámico del modelo. La etapa más compleja de la construcción.
- *Agent*: Define las entidades. Tiene *Agent Components* (atributos, variables, eventos) y *Statechart* (máquina de estado). Los agentes pueden ser entidades auxiliares que realizan acciones que no necesariamente tienen una correspondencia con el sistema real (por ejemplo, realizan cálculos).
- *Analysis*: Para recolección y análisis de resultados. Tiene *Charts* (gráficos, se visualizan durante la ejecución) y *Data* (se visualizan y también sirven para alimentar gráficos, por ejemplo).
- *Controls*: Para controlar interactivamente la simulación.

## **AnyLogic: construcción del modelo (en detalle)**

Los elementos del modelo se deben **conectar en espacio y tiempo**.

Una secuencia común de pasos es la siguiente (algunas propiedades de los componentes tienen valores por defecto, otras se deben configurar):

**Etapas 1.** Construir la estructura estática desde *Space Markup* a partir de *Rectangular Node* y *Path*.

## AnyLogic: construcción del modelo (en detalle)

**Etapas 2.** Especificar los componentes dinámicos (algunos se deben ligar a componentes estáticos) desde *Process Modelling Library/Blocks*, ejemplos:

- *Source*: Ingreso al modelo. Las propiedades incluyen el tiempo entre arribos, agente y ubicación (de la estructura estática) a la que arriba.
- *Queue*: Cola de espera. Las propiedades incluyen agente, ubicación estática, capacidad, política.
- *Delay*: Demora por actividad. Las propiedades incluyen duración, capacidad, ubicación estática, agente.
- *Sink*: Salida del modelo. Las propiedades incluyen al agente.

## AnyLogic: modelado de recursos

Diferentes alternativas para modelar una dinámica de interacción entre una o varias colas de entidades y uno o varios servidores que actúan como recursos que restringen el comienzo de una actividad:

- *Source – Queue – Delay – Sink*: La capacidad del recurso se especifica en las propiedades del bloque *Delay*.
- *Source – Seize – Delay – Release – Sink* (con *Resource Pool*): Es la alternativa más flexible. En los bloques *Seize/Release* se toma/libera el recurso. Los recursos se modelan con un bloque *Resource Pool*.
- *Source – Service – Sink* (con *Resource Pool*): Es la alternativa más compacta. El bloque *Service* concentra la mayoría de la lógica.

Más detalles: *The Art of Process-Centric Modeling with AnyLogic*, p. 34.



## AnyLogic: otros elementos específicos de modelado

- Bifurcaciones de flujos: Bloques *Select Output* y *Select Output5*.
- Prioridades de acceso a recursos: Bloque *Seize* permite configurar prioridades para acceder a un *Resource Pool*. Ver *The Art of Process-Centric Modeling with AnyLogic*, p. 179.

Cuando se configuran las propiedades de los diferentes bloques, se pueden especificar valores fijos o se puede acceder a propiedades de los componentes del modelo, accesibles a través de Java (especie de API dinámica del modelo). Si se presiona las teclas Control+Espacio en un cuadro de propiedades, se despliega la lista de propiedades (variables, operaciones) que se pueden incluir.

## AnyLogic: ejecución del modelo

Mediante el botón *Build model (F7)* se construye automáticamente el modelo (se genera el programa ejecutable). La construcción implica un chequeo de consistencia previo. Si hay errores, se indica en la salida y el modelo no se puede ejecutar hasta que no sea consistente.

Mediante el botón *Run (F5)* se ejecuta el modelo, una vez chequeado y construido. La ejecución proporciona una salida visual por defecto, derivada del diagrama de estructura dinámica, que incluye algunas métricas en tiempo real que permiten realizar una primera verificación del modelo. La ejecución del modelo es computacionalmente demandante.

## **AnyLogic: otras funcionalidades**

En el Tema 11 retomaremos algunas funcionalidades de AnyLogic, en particular aquellas para recolección y visualización de resultados (*Analysis/Data*, *Analysis/Charts*, bloques *Time Measure Start* y *Time Measure End*).