

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel

SimJava

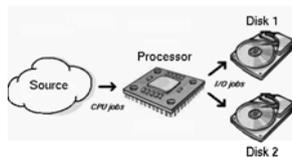
Lenguajes de Simulación
Curso 2009/2010

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel
[Introducción a SimJava](#)

- Paquete de simulación sobre Java.
- Entidades ejecutándose en diferentes hilos.
- Interconectadas por sus puertos: eventos.
- Clase Principal (*gestora*):
 - Hilos de simulación.
 - Avance de tiempo.
 - Eventos.
- Registro de la simulación en un fichero de traza.

Lenguajes de Simulación - Curso 2009/2010

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel
[Conceptos Fundamentales](#)



- Procesador
- 2 Discos: probabilidad de uso
- Analizar el efecto de sustituir un componente.

Lenguajes de Simulación - Curso 2009/2010

■ **sim_entity**

- Encapsula la funcionalidad disponible para entidades.
- Nueva entidad: **extends sim_entity**
 - Definición de puertos.
 - Propiedades específicas.
 - Estadísticas.
- Implementación del funcionamiento: **body()**
- Puertos:
 - Definición: **sim_port**
 - Adición: **add_port()**
 - Un puerto tiene un único puerto destino

```
import eduni.simjava.*; // Import the SimJava basic package

class Source extends Sim_entity {
    private Sim_port out;

    Source(String name) {
        super(name); // Call the constructor of Sim_entity
        out = new Sim_port("Out"); // Create the port
        add_port(out); // Add the port to the entity
    }

    ...
}
```

■ **body()**

- Sobreescritura para la caracterización de la entidad.
- Generación/recepción de eventos (emisor/receptor).
 - Colas de eventos futuros: entrada / salida.
 - Gestión de eventos en todo momento (incluso con la entidad en estado "ocupado").

■ Gestión de eventos:

- Envío de eventos: **sim_schedule()**
 - Identificador del evento.
 - Destinatario del evento (entidad + puerto).
 - Retardo para la entrega del evento.
- Recepción de eventos: **sim_event()**
 - Datos y parámetros disponibles en el receptor.

■ Espera de eventos futuros:

- Espera indefinida: **sim_wait()**
- Espera condicionada: **sim_event_for()**
 - Eventos de una determinada naturaleza.
 - Eventos en un determinado momento.

■ Comprobación de colas:

- Elementos en la cola: **sim_waiting()**
- Seleccionar primer elemento: **sim_select()**
- **sim_get_next()**
 - **sim_waiting() + sim_select() + sim_wait()**
 - Dos variantes:
 - Si no hay evento, sigue esperando.
 - Si no hay un evento etiquetado, sigue esperando.

■ Estado de Procesamiento:

- Ocupada un cierto tiempo: `sim_process()`
- `sim_process_for()`
 - Durante un determinado tiempo (cualquier evento puede detener el procesamiento).
 - Hasta la llegada de un determinado evento (aquel que coincide con el predicado)
- `sim_process_until()`
 - Espera hasta una determinada llegada (Tiempo ilimitado)
 - Evento.
 - Evento etiquetado.

■ Otros:

- `sim_cancel()`
 - Elimina el primer elemento enviado de la cola de futuros.
- `sim_completed()`
 - Indica que el procesamiento de un evento se ha finalizado.

- *Source* genera 100 eventos:
 - Envía al procesador un evento cada 50 udsT.
- *Procesador* necesita 30 udsT para el servicio.
- *Disco 1* necesita 60 udsT para el servicio.
- *Disco 2* necesita 110 udsT para el servicio.

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel

Ejemplo

```
class Source extends Sim_entity {
private Sim_port out;
private double delay;

Source(String name, double delay) {
super(name);
this.delay = delay;
out = new Sim_port("Out");
add_port(out);
}

public void body() {
for (int i=0; i < 100; i++) {
sim_schedule(out, 0.0, 0);
sim_pause(delay);
}
}
}
```

Lenguajes de Simulación - Curso 2009/2010

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel

Ejemplo

```
class Sink extends Sim_entity {
private Sim_port in, out1, out2;
private double delay;

Sink(String name, double delay) {
super(name);
this.delay = delay;

// Port for receiving events from the source
in = new Sim_port("In");
// Port for sending events to disk 1
out1 = new Sim_port("Out1");
// Port for sending events to disk 2
out2 = new Sim_port("Out2");

add_port(in);
add_port(out1);
add_port(out2);
}
}
```

Lenguajes de Simulación - Curso 2009/2010

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel

Ejemplo

```
public void body() {
int i = 0;

while (Sim_system.running()) {
Sim_event e = new Sim_event();
// Get the next event
sim_get_next(e);
// Process the event
sim_process(delay);
// The event has completed service
sim_completed(e);

if ((i % 2) == 0) {
sim_schedule(out1, 0.0, 1);
} else {
sim_schedule(out2, 0.0, 1);
}
i++;
}
}
```

Lenguajes de Simulación - Curso 2009/2010

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel

Ejemplo

```
class Disk extends Sim_entity {
    private Sim_port in;
    private double delay;

    Disk(String name, double delay) {
        super(name);
        this.delay = delay;
        // Port for receiving events from the processor
        in = new Sim_port("In");
        add_port(in);
    }

    public void body() {
        while (Sim_system.running()) {
            Sim_event e = new Sim_event();
            // Get the next event
            sim_get_next(e);
            // Process the event
            sim_process(delay);
            // The event has completed service
            sim_completed(e);
        }
    }
}
```

Lenguajes de Simulación - Curso 2009/2010

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel

Simulando

- Gestionada por `sim_system()`
 - Inicialización.
 - Instanciar cada entidad.
 - Enlazar los puertos de las entidades.
 - Ejecutar la simulación.

Lenguajes de Simulación - Curso 2009/2010

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel

Simulando

```
public static void main(String[] args) {
    // Initialise Sim_system
    Sim_system.initialise();

    // Add the source
    Source source = new Source("Source", 50);

    // Add the processor
    Sink processor = new Sink("Processor", 30);

    // Add disks
    Disk disk1 = new Disk("Disk1", 60);
    Disk disk2 = new Disk("Disk2", 110);

    // Link the entities' ports
    Sim_system.link_ports("Source", "Out", "Processor", "In");
    Sim_system.link_ports("Processor", "Out1", "Disk1", "In");
    Sim_system.link_ports("Processor", "Out2", "Disk2", "In");

    // Run the simulation
    Sim_system.run();
}
```

Lenguajes de Simulación - Curso 2009/2010

- Selección condicionada de eventos.
- `sim_predicate()`
 - Predicados generales.
 - Seleccionar cualquier evento.
 - No seleccionar ningún evento.
 - Predicados sobre el evento etiquetado.
 - Seleccionar/no evento con una etiqueta dada.
 - Predicados sobre la entidad emisora.
 - Seleccionar/no evento procedente de una entidad emisora.

- La simulación es un programa complejo
 - Trazo de *debugging*.
 - Validación del modelo.
- `sim_trace()`
 - Trazo “por defecto”.
 - Trazo por entidad.
 - Trazo por evento.

- Definición de tiempos de servicio/pausa.
- Generador de números aleatorios: $U(0,1)$.
 - Pseudoaleatorios: replicabilidad de la serie.
- `eduni.simjava.distributions`
 - Paquete con 22 distribuciones.
 - Instancia directa de la distribución: `sample()`.
 - Inicializadas con una raíz específica.

- Medidas estadísticas para la evaluación del rendimiento de las entidades.
- ¿Qué medidas nos interesan?
 - Medidas basadas en tasas.
 - Ocurrencias de un evento en un determinado periodo.
 - Medidas basadas en estado (continuas/no continuas)
 - La entidad pasa, o no, continuamente de un estado a otro.
 - Medidas basadas en intervalo.

- Basadas en tasas:
 - Tasa de llegadas.
 - Rendimiento.
- Basadas en estado:
 - Longitud de la cola.
 - Utilización.
- Basadas en intervalo:
 - Tiempo de espera.
 - Tiempo de servicio.
 - Tiempo de residencia.

- Añadir medidas estadísticas a las entidades.
- `add_measure()` + `sim_stat()`
 - Constantes con las medidas disponibles.
- Medidas calculadas al final de la simulación.
- Medidas para eventos etiquetados.

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel
Definiendo Condiciones de Finalización

■ Definición de un estacionario:

`set_transient_condition()`

- Eventos completados.
- Tiempo transcurrido.
- Método máximo/mínimo (basado en medidas)
- Ninguno: interesa estudiar el estacionario

Lenguajes de Simulación - Curso 2009/2010

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel
Definiendo Condiciones de Finalización

■ Definición de condiciones de terminación:

`set_termination_condition()`

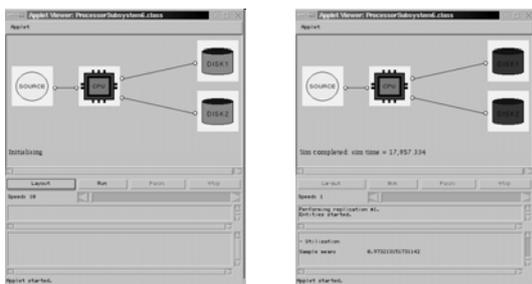
- Eventos completados.
- Tiempo transcurrido.
- Condiciones de terminación basadas en intervalo.
- Ninguno: "simulación infinita".

Lenguajes de Simulación - Curso 2009/2010

Librerías Orientadas a la Simulación para su uso en Lenguajes de Alto Nivel
Animación

■ Simulación animada basada en *applets*.

- Facilita la validación del modelo.



Lenguajes de Simulación - Curso 2009/2010

- Tutorial SimJava
 - <http://www.icsa.inf.ed.ac.uk/research/groups/hase/simjava/guide/tutorial.html>
- Página Oficial SimJava
 - <http://www.dcs.ed.ac.uk/home/hase/simjava/>
