Herramientas para Simulación a Eventos Discretos

Clase nro. 15

Objetivo

- Mostrar la aplicación de conceptos presentados en el curso, en herramientas alternativas para Simulación a Eventos Discretos:
  - DesmoJ: Biblioteca de programación de simulaciones.
  - ProModel: Entorno de construcción de simulaciones.

Herramientas de simulación

- Anteriormente se presentaron las facilidades provistas por las herramientas de software de simulación.
- Banks *et al* (2001) propone una clasificación de herramientas según:
  - Lenguajes de programación de propósito general (por ejemplo C++).
  - Lenguajes de programación de simulaciones (por ejemplo DesmoJ).
  - Paquetes de simulación (por ejemplo ProModel).

DesmoJ

- DesmoJ es una biblioteca open source para simulación a eventos discretos.
- Es desarrollada por la Universidad de Hamburgo y está implementada en Java.
- Soporta el enfoque de interacción de procesos y eventos discretos en 2 fases.
- Brinda varias facilidades para el desarrollo de SED.

DesmoJ

Modelado:

- Model
  - Es el modelo a simular, puede contener submodelos.
  - Operaciones abstractas: init(), doInitialSchedules()
- Entity
  - Son las entidades de un modelo.
  - Poseen una prioridad numérica.

DesmoJ

```
import desmoj.core.simulator.*;
import desmoj.core.dist.*;

public class HospitalModel extends Model {
  public ProcessQueue pacientesEsperando;
  public RealDistExponential arribos, estancias;
  private int camas, cantCamas;
  private double mediaLlegadas, mediaEstancia;

  public  HospitalModel(Model owner, double mediaLlegadas, double mediaEstancia,
    int cantCamas, boolean showInReport, boolean showInTrace) {
    super(owner, "HospitalModel", showInReport, showInTrace);
    this.mediaLlegadas = mediaLlegadas;
    this.mediaEstancia = mediaEstancia;
    this.cantCamas = cantCamas;
  }

  public void init() {
    pacientesEsperando = new ProcessQueue(this, "Pacientes Esperando", true, true);
    arribos = new RealDistExponential(this, "Arribos", mediaLlegadas, true, true);
    estancias = new RealDistExponential(this, "Estancias", mediaEstancia, true, true);
    camas = cantCamas;
  }
```

DesmoJ

```
  public String description() {
    return "Hospital Simple";
  }

  public boolean hayCamasLibres() {
    return (camas > 0);
  }

  public void tomarCama() {
    camas--;
  }

  public void liberarCama() {
    camas++;
  }

  public void doInitialSchedules() {
    PacienteFeeder feeder = new PacienteFeeder(this, true);
    feeder.schedule(new SimTime(0.0));
  }
}
```

DesmoJ

Modelado:

- Event
  - Pueden ser Externos o Internos.
  - Se crean dinámicamente a lo largo de la simulación.
  - Operaciones abstractas: eventRoutine() o eventRoutine(Entity)

DesmoJ

```
import desmoj.core.simulator.*;

public class PacienteFeeder extends ExternalEvent {
  private int cuantos;

  public PacienteFeeder(Model owner, boolean showInTrace) {
    super(owner, "PacienteFeeder", showInTrace);
    cuantos = 0;
  }

  public void eventRoutine() {
    cuantos++;
    HospitalModel model = (HospitalModel)getModel();
    Paciente pac = new Paciente(model, "Paciente", cuantos, true);
    pac.activate(new SimTime(0.0));
    schedule (new SimTime(model.arribos.sample()));
  }

}
```

DesmoJ

Modelado:

- SimProcess
  - Los procesos modelan el ciclo de vida de una entidad.
  - Están implementados sobre una thread (aúnque no hay varias threads ejecutando a la vez).
  - Hereda de Entity.
  - Operaciones Abstractas: lifeCycle()
  - Primitivas: passivate(), activate(), hold(SimTime)

DesmoJ

```
import desmoj.core.simulator.*;

public class Paciente extends SimProcess {
  private HospitalModel myModel;
  private int id;

  public Paciente(Model owner, String name, int id, boolean showInTrace) {
    super(owner, name, showInTrace);
    this.id = id;
    myModel = (HospitalModel)owner;
  }

  public void lifeCycle() {
  sendTraceNote("El paciente " + id + " llego al hospital.");
    if (!myModel.hayCamasLibres()) {
      myModel.pacientesEsperando.insert(this);
      passivate();
    }
    myModel.tomarCama();
    sendTraceNote("El paciente " + id + " tomo una cama.");
    hold(new SimTime(myModel.estancias.sample()));
    myModel.liberarCama();
    if (myModel.pacientesEsperando.length() > 0) {
      Paciente pac = (Paciente) myModel.pacientesEsperando.first();
      myModel.pacientesEsperando.remove(pac);
      pac.activateAfter(this);
    }
    sendTraceNote("El paciente " + id + " se retira.");
  }
}
```

DesmoJ

Corridas:

- Experiment:
  - Esta clase corre modelos.
  - Posee un calendario (Scheduler) y manejador de streams (DistributionManager).
  - Al terminar cada corrida genera un reporte, un tracer, un archivo de errores y uno de debug.
- Scheduler:
  - Es un calendario mixto de eventos y procesos.
  - Capaz de correr modelos híbridos.

DesmoJ

```
public static void main (String args[]) {

  HospitalModel hospital = new HospitalModel(null, 6, 60, 20, true, true);

  // creo un experiment
  Experiment exp = new Experiment("HospitalSimple");
  // conecto el modelo con el experimento
  hospital.connectToExperiment(exp);

  // preparo la simulacion
  exp.setShowProgressBar(true);
  exp.stop(new SimTime(24 * 30));
  // activo el tracer
  exp.tracePeriod(new SimTime(0.0), new SimTime(24 * 30));

  // comienzo la simulacion
  exp.start();

  // creo el report
  exp.report();
  // termino el experimento
  exp.finish();
}
```

Otras bibliotecas

- C++Sim (Interacción de Procesos)
- Simpy (Interacción de Procesos)
- Simkit (Grafos de Eventos)
- Adevs (Formalismo Devs)

ProModel

- Paquete de simulación a eventos discretos, orientado a sistemas de manufactura.
- Provee la gran mayoría de las facilidades para la construcción de modelos de SED, y para la experimentación con los mismos.
- Ejemplo: hospital simple.

ProModel

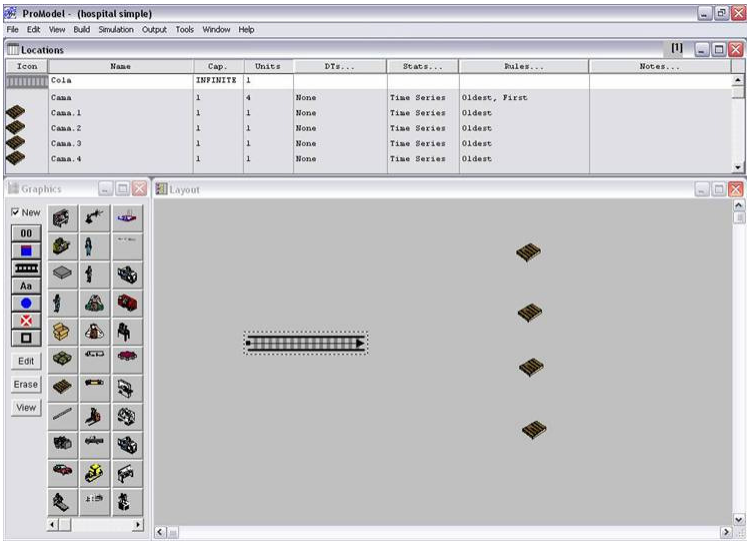
- El modelo se construye gráficamente, utilizando componentes predefinidos.
- Elementos básicos de modelado:
  - Locations
  - Entities
  - Arrivals
  - Processing

ProModel

Locations:

- Ubicaciones en el espacio.
- Propiedades:
  - Icono, nombre.
  - Capacidad: máximo número de entidades a la vez en la ubicación.
  - Unidades: modela varias locations idénticas.
  - Downtimes, estadísticas.
  - Reglas de entrada y de salida.

ProModel

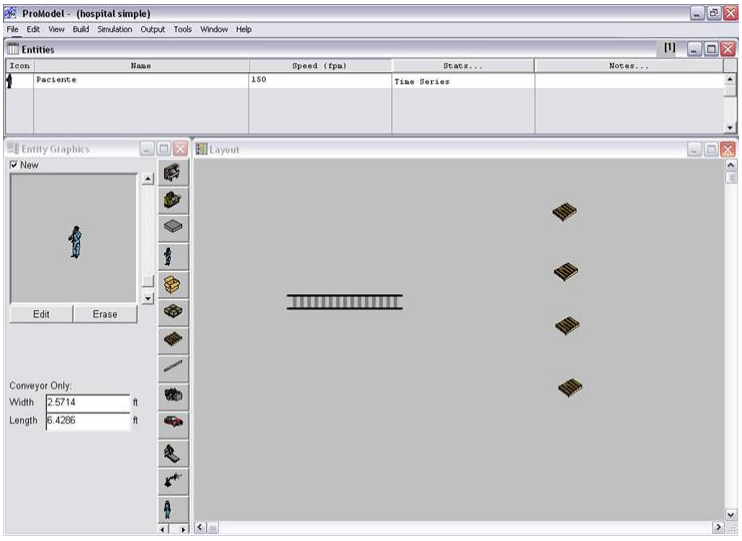


ProModel

Entities:

- Entidades que se mueven entre locations.
- Propiedades:
  - Icono, nombre.
  - Velocidad: se utiliza para calcular tiempos de desplazamiento.
  - Estadísticas.

ProModel

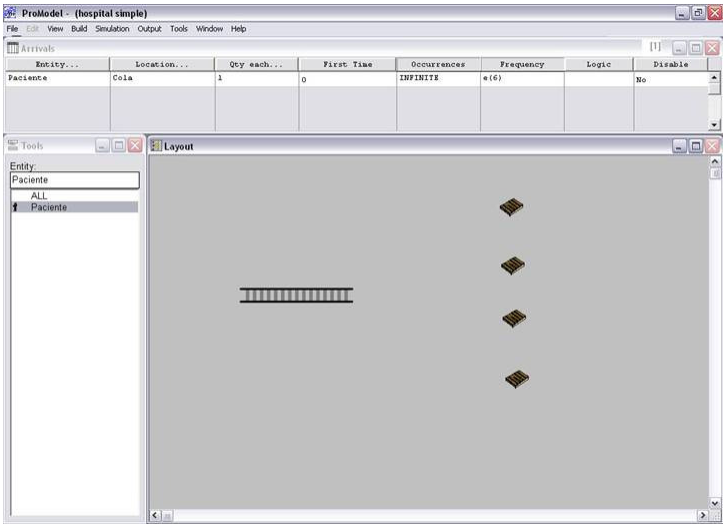


ProModel

Arrivals:

- Feeders de entidades.
- Propiedades:
  - Entity, location.
  - Quantity: cantidad de entidades en cada arribo.
  - First time, ocurrences.
  - Frequency: tiempo entre arribos.
  - Logic: código que ejecuta en cada arribo.
  - Disable.

ProModel

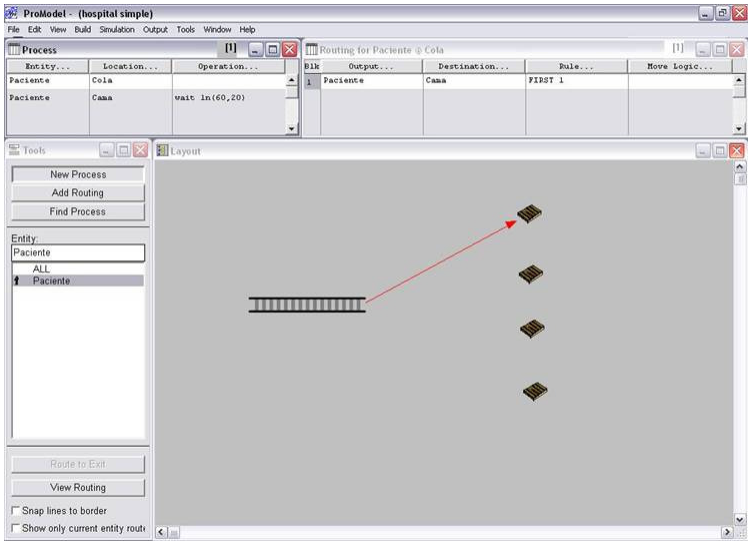


ProModel

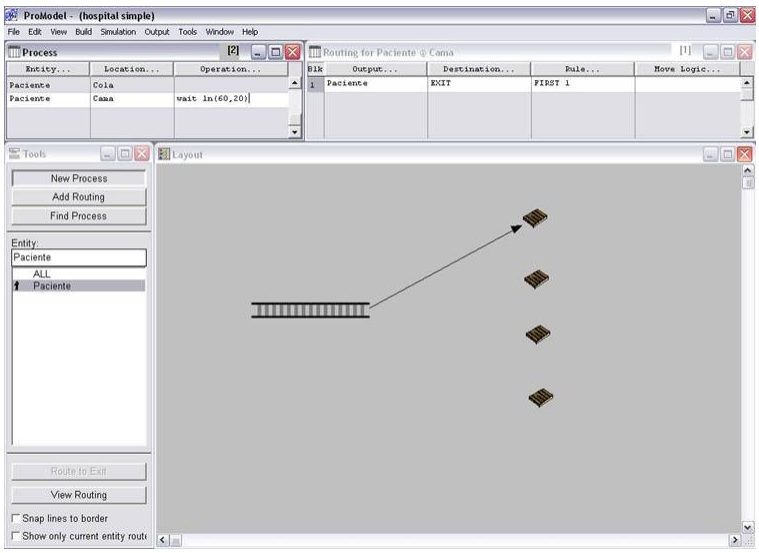
Processing:

- Define las trayectorias de las entidades y sus tiempos de permanencia en las locations.
- Process: actividad de una entity en una location (propiedades: entity, location, operation).
- Routing: define hacia donde va una entity luego de finalizar una actividad (propiedades: output, destination, rule, move logic).

ProModel



ProModel



ProModel

Otras facilidades:

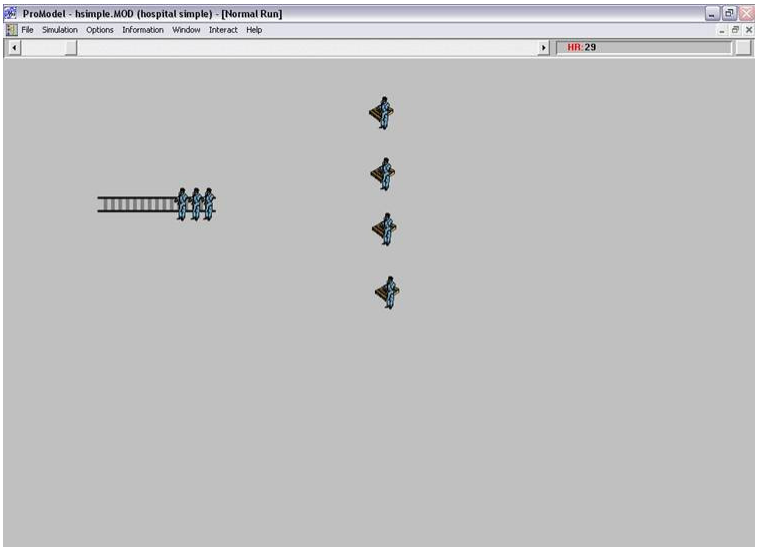
- Atributos.
- Variables.
- Distribuciones empíricas.
- Path network: para definir desplazamientos de entidades.

ProModel

Ejecución del modelo:

- Varias replicaciones.
- Tiempo de estado estacionario (especificado por el usuario).
- Recolección automática de estadísticas.
- Herramienta de análisis de resultados.

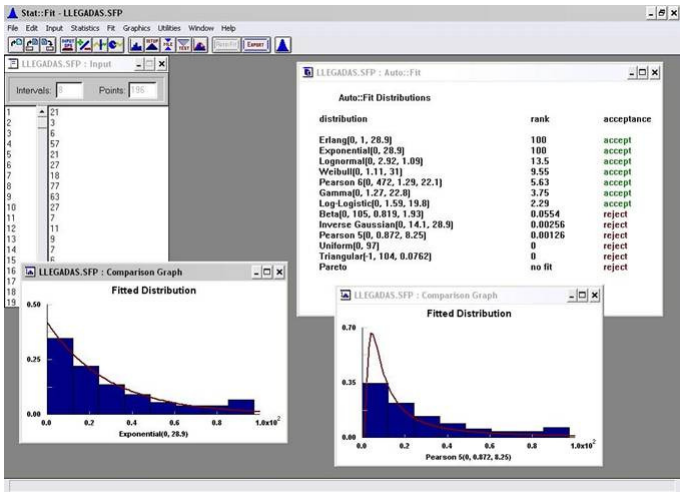
ProModel



ProModel

- Herramientas relacionadas:
  - Stat::Fit: Preparación de los datos de entrada, análisis de datos de salida.
  - SimRunner: Optimización de funciones de costo, según variables de decisión, utilizando Algoritmos Genéticos.
  - Service Model: Paquete para modelado de sistemas de servicio (por ejemplo un banco).
- Sitio web: www.promodel.com

ProModel



Otros paquetes

- Arena
- Automod
- Extend
- Micro Saint
- WITNESS

Bibliografía

Discrete-event system simulation. Banks, J., Carson, J., Nelson, B., Nicol, D. (2001) Prentice Hall.

Simulation using ProModel. Harrel, Ch., Ghosh, B., Bowden, R. (2000) Mc Graw Hill

DesmoJ home page. http://www.desmoj.de/