

Programación Orientada a Objetos

Francisco Hidrobo / Kay Tucci / Mayerlin Uzcátegui

Semestre A 2009

Proyectos

Galatea es una plataforma de simulación para modelos de sistemas multi-agentes. Este es un objetivo subsidiario de varias metas de investigación del Centro de Simulación y Modelos de la ULA (CeSiMo). Galatea está basada en Glider, una plataforma también diseñada en la ULA y que ha sido usada en Venezuela en diversos proyectos e instituciones. Galatea está siendo desarrollada usando POO.

1. Autómatas celulares

Un Autómata Celular (A.C.) es un sistema dinámico que evoluciona en pasos discretos. Es adecuado para modelar sistemas naturales que puedan ser descritos como una colección masiva de objetos simples que interactúen localmente unos con otros.

Los autómatas celulares pueden ser usados para modelar numerosos sistemas físicos que se caractericen por un gran número de componentes homogéneos y que interactúen localmente entre sí. De hecho, cualquier sistema real al que se le puedan analogar los conceptos de “vecindad”, “estados de los componentes” y “función de transición” es candidato para ser modelado por un A.C.

Las características de los autómatas celulares harán que dichos modelos sean discretos en tiempo, espacio o ambos (dependiendo de la variante de la definición de A.C. que se use). Algunos ejemplos de áreas en donde se utilizan los autómatas celulares son:

- Modelación de flujo de tránsito vehicular y de peatones.
- Modelación de fluidos (gases o líquidos).
- Modelación de la evolución de células o virus como el VIH.
- Modelación de procesos de precolación.

No existe una definición formal y matemática aceptada de Autómata Celular, sin embargo, se puede describir a un A.C. como tuplas cuyos elementos se describen a continuación.

Un A.C. consiste de:

- Una rejilla, reticulado o malla de enteros infinitamente extendida, y con dimensión d entera. Cada celda de la malla se conoce como célula.

- Cada celda puede tomar un conjunto de valores enteros a partir de un conjunto finito de estados k .
- Cada celda además se caracteriza por su vecindad, un conjunto finito de celdas en las cercanías de la misma.
- De acuerdo con esto, se aplica a todas las celdas de la cuadrícula, una función de transición (f) que toma como argumentos los valores de la celda en cuestión y los valores de sus vecinos, y regresa el nuevo valor que la celda tendrá en la siguiente etapa de tiempo. Esta función f se aplica, como ya se dijo, de forma homogénea a todas las celdas, por cada paso discreto de tiempo.”¹

Se quiere desarrollar una pieza de software que permita simular y visualizar la evolución de un autómata celular cuyas regla, condiciones iniciales y vecindad sean definidas por el modelista.

2. Dinámica de sistemas

Uno de los paradigmas clásicos de simulación de sistemas dinámicos es conocido como dinámica de sistemas, donde se utilizan los conceptos básicos de niveles y flujos para modelar sistemas complejos donde el transcurso de tiempo y el espacio de estados son continuos. Este paradigma generalmente permite simular el comportamiento del sistema y detectar problemas prácticos con la antelación que se necesita para intervenirlos.

La dinámica de sistemas trata la aproximación a la modelización de la dinámica de sistemas complejos, tales como las galaxias, poblaciones biológicas o los sistemas económicos, en los que las partes demuestran propiedades interactivas entre los objetos, dando como resultado una comunicación gracias a las transacciones energéticas que se derivan de las relaciones mutuas.

Fue fundada a principios de la década de 1960 por Jay Forrester, de la MIT Sloan School of Management (Escuela de Administración Sloan, del Instituto Tecnológico de Massachusetts) con el establecimiento del MIT System Dynamics Group (Grupo de dinámica de sistemas del MIT). En esa época había empezado a aplicar lo que había aprendido con sistemas eléctricos a toda clase de sistemas.

Lo que distingue a la dinámica de sistemas de otras aproximaciones al estudio de los mismos problemas, es el uso de bucles de realimentación. La construcción de modelos por la dinámica de sistemas parte de la distinción esencial entre niveles y flujos. Con estos elementos se edifican modelos que describen sistemas internamente conectados por bucles de realimentación; sistemas que muestran el carácter no lineal que se suele encontrar en los problemas del mundo real. ¹

La formulación de un modelo en dinámica de sistemas consta de 4 pasos que pueden ser resumidos en las siguientes etapas:

Conceptualización:

- Definir el propósito del modelo
- Definir las fronteras del modelo

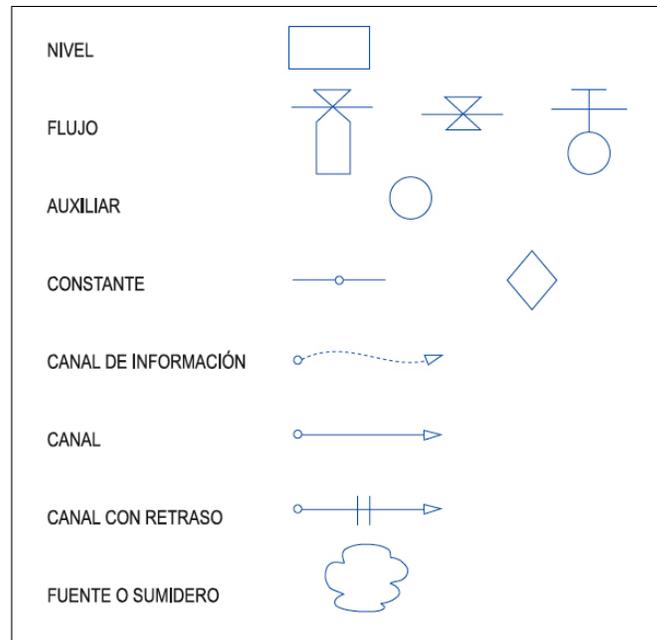


Figura 1: Símbolos utilizados en el Diagrama de Forrester

- Describir el comportamiento a través de diagramas de referencia de las variables principales
- Diagramar los mecanismos básicos, ciclos de realimentación, del sistema

Formulación:

- Convertir los diagramas de retroalimentación en ecuaciones de flujo y almacenamiento
- Estimar y seleccionar el valor de los parámetros

Prueba:

- Simular el modelo y probar las hipótesis dinámicas
- Probar los supuestos del modelo
- Comprobar el comportamiento del modelo y la sensibilidad a perturbaciones

Implementación:

- Probar las respuestas del modelo a las diferentes políticas
- Traducir los resultados del estudio para hacerlos accesibles

Se requiere desarrollar una interfaz gráfica que permita modelar a partir de la notación gráfica de dinámica de sistemas (basada en niveles y flujos) genere el código del nodo continuo equivalente en Galatea, Esta interfaz permitirá al modelista llevar a cabo las tareas de Conceptualización y Formulación utilizando los símbolos de diagrama mostrados en la figura 1.

3. Animador de la traza de Galatea

Uno de los módulos que aún no se han desarrollado en Galatea es el animador de la traza de simulación.

Con este proyecto se busca desarrollar un prototipo de animador de traza para alguno de los modelos ya probados sobre la plataforma Galatea. El animador debe leer un archivo de traza de una simulación y con los datos contenidos en él construir la red de nodos, y la lista de eventos que se llevaron a cabo durante la simulación. A partir de esta información debe animar el envío de mensajes entre los nodos del sistema.

En el archivo de traza (`.trc`) encontramos una marca de inicio de evento de la forma

```
>>>>> 2 10.0 Entrada[1] FEL=[1;1;/] -(R)Taquilla[1],45.0
```

donde podemos notar los siguientes campos

- marca de evento
- numero del evento
- tiempo de simulacion
- nodo asociado
- lista de eventos futuros

Dentro de cada evento estan marcados los procesos macro de activacion del nodo

```
10.0 Act (I)Entrada[1]
                                     ...
10.0 (I)Entrada[1] Node ***->sendto() Entrada1:2 ->Taquilla[2]
```

En este caso en el tiempo 10 entre otras cosas ocurre la activación del nodo tipo Input denominado *Entrada*[1], eso provoca que se envíe el mensaje 2 que genero el nodo *Entrada*[1] al nodo *Taquilla*[2].

Posteriormente ocurre la revision de los nodos restantes

```
10.0 Scan (R)Taquilla[2]
10.0 (R)Taquilla[2] Node ***->act() 55.0
```

en el mismo tiempo 10 durante la revisión del nodo *Taquilla*[2] se detecta que es necesario activar el nodo *Taquilla*[2] en el tiempo 55

Esto se realiza con las respectivas llamadas a los metodos básicos durante estas activaciones o revisiones.

Cada nodo tendrá una representación gráfica en la animación y se podrán mover de tal forma que el usuario los pueda disponer de la forma más conveniente. Los mensajes, que también tendrán una representación gráfica de acuerdo a su tipo, se moverán entre los nodos sobre un camino definido como una polilínea.

El usuario debe poder cambiar la velocidad de la animación desde paso a paso hasta un evento por segundo, así como también debe poder mostrar detalles de algún sector de la red que sea de su interés; mostrando en todo momento el nivel en el archivo de traza por donde va la animación y el tiempo de simulación y la ubicación en la lista de eventos.

tarea 1

Dado uno de los proyectos de la sección anterior y realice las siguientes actividades:

- Identifique los posibles objetos del sistema.
- Cuáles de los posibles objetos son relevantes en el sistema y cuáles no lo son. Explique el por qué.
- De los objetos relevantes al problema diga cuáles son sus atributos y sus métodos, indicando si son públicos o privados.

tarea 2

Utilizando los resultados de las correcciones de la tarea 1, realice las siguientes actividades:

- Identifique las clases
- Generalice las clases para obtener superclases
- Cree la jerarquía de clases del problema
- Declare cada una de las clases con sus atributos y métodos sin implementarlos.