Resumen Modelado y Simulación

24-04-2024

Objetivos docentes

Una vez estudiado el contenido del tema y realizados los ejercicios prácticos, debería saber:

- Discutir el significado de los conceptos “sistema”, “modelo”, “experimento” y “simulación”.
- Discutir en qué situaciones puede ser imposible o desaconsejable experimentar con el sistema real.
- Describir y comparar las diferentes formas de estudiar un sistema y los diferentes tipos de modelos.
- Comparar y reconocer los distintos tipos de modelos matemáticos.
- Discutir los niveles en el conocimiento de los sistemas de la clasificación de Klir, y el significado de los conceptos “análisis”, “inferencia” y “diseño” de sistemas con relación a dicha clasificación del conocimiento.
- Discutir un marco formal para el modelado y la simulación en el que se definen 5 entidades (sistema fuente, base de datos del comportamiento, modelo, simulador y marco experimental) y dos relaciones entre ellas (relación de modelado y relación de simulación).
- Describir las diferencias entre los distintos tipos de modelos matemáticos y saber reconocer a qué tipo pertenece cualquier modelo.
- Discutir qué características debe tener un modelo para ser de tiempo discreto y cómo se realiza la descripción y simulación de este tipo de modelos.
- Discutir qué es un autómata celular y cómo se realiza su descripción y simulación.
- Discutir qué características tienen los modelos de eventos discretos y las diferencias entre su descripción orientada a la planificación de eventos y orientada a los procesos.
- Discutir los pasos de que típicamente consta un estudio de simulación en el cual se emplean modelos estocásticos de eventos discretos.
- Emplear el lenguaje R para realizar programas sencillos y representaciones gráficas sencillas de los datos.

Introducción

El texto aborda los fundamentos del modelado y la simulación, comenzando por la definición de conceptos clave como modelo, sistema, experimento y simulación. Se presenta una clasificación en cuatro niveles del conocimiento sobre un sistema y se destaca la importancia de la especificación del conocimiento mediante modelos matemáticos. Se describe un marco formal para el modelado y la simulación, incluyendo entidades como el sistema fuente, el modelo y el simulador, así como la relación entre ellas. Se profundiza en la especificación del conocimiento, explorando clasificaciones de modelos matemáticos, especialmente modelos de tiempo discreto y eventos discretos. También se detallan los pasos típicos de un estudio de simulación con modelos estocásticos de eventos discretos, desde la definición del problema hasta la documentación de conclusiones. Se introduce el lenguaje R como una herramienta esencial para el análisis de datos en el contexto del modelado matemático y la simulación.

Conceptos fundamentales

Sistema, experimento y modelo

El modelado matemático y la simulación por computadora son herramientas poderosas para comprender el comportamiento de los sistemas. Un sistema puede ser cualquier conjunto de elementos interrelacionados que funcionan juntos, como una fábrica, un hospital, una red de computadoras o un parque temático.

Una forma de conocer el comportamiento de un sistema es experimentar con él directamente. Sin embargo, esto no siempre es posible o viable, debido a razones como:

- No existencia del sistema real: En la fase de diseño de nuevos sistemas, es necesario predecir su comportamiento antes de construirlos.
- Escala de tiempo: En estudios geológicos, cosmológicos o de evolución, los experimentos deberían durar miles o millones de años.
- Costo elevado: Experimentar con el sistema real puede ser muy costoso, como en el caso de ampliar una fábrica.
- Peligro o perjuicio: Experimentar con el sistema real puede ser peligroso o perjudicial, como probar un nuevo sistema de facturación en un aeropuerto.

Una alternativa a la experimentación con el sistema real es construir un modelo del sistema y experimentar con él. Un modelo es una representación simplificada del sistema que permite estudiar su comportamiento sin necesidad de hacerlo en el sistema real. Existen diferentes tipos de modelos:

- Modelos mentales: Representaciones internas del sistema en la mente del observador.
- Modelos verbales: Descripciones del sistema usando palabras.
- Modelos físicos: Réplicas físicas del sistema a escala reducida.
- Modelos matemáticos: Representaciones del sistema mediante ecuaciones, algoritmos o estructuras matemáticas.

Los modelos matemáticos son particularmente útiles para estudiar el comportamiento de sistemas complejos. En algunos casos, las ecuaciones del modelo pueden resolverse analíticamente, pero en la mayoría de los casos se requiere simulación por computadora. La simulación consiste en utilizar un programa de computadora para imitar el comportamiento del sistema descrito por el modelo.

Este texto se centrará en los modelos matemáticos y su resolución mediante simulación por computadora. Los modelos mentales, verbales y físicos no se abordarán en profundidad. Se utilizará el término “modelo” para referirse a “modelo matemático”. La resolución analítica de modelos matemáticos también está fuera del alcance de este texto.

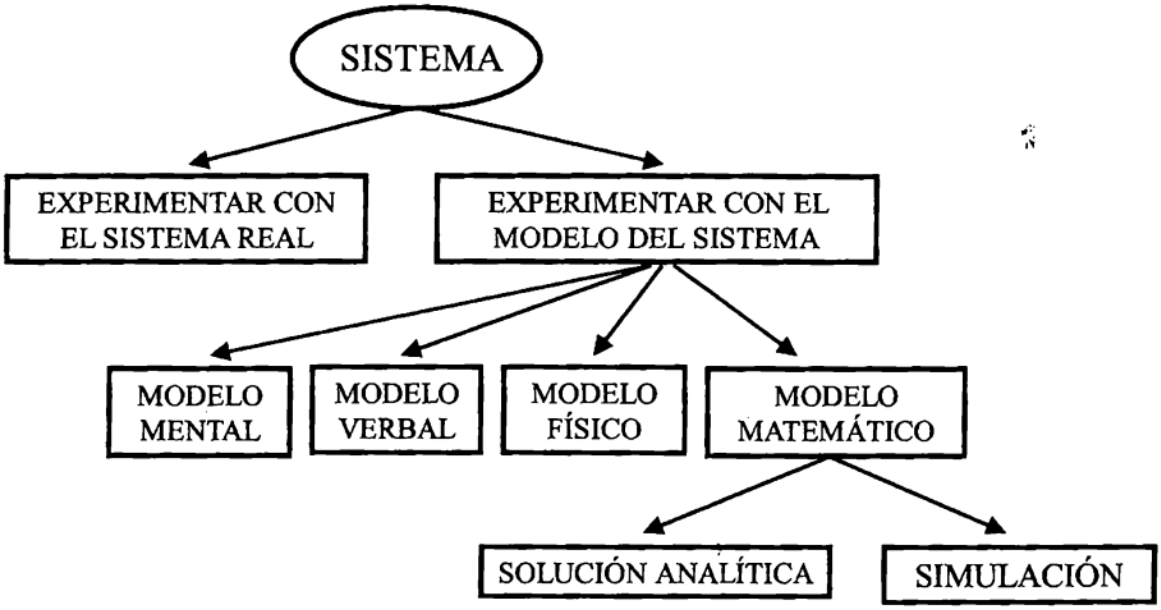

Figura 1.1: Formas de estudiar un sistema.

Figure 1: Imagen

Niveles en el conocimiento de los sistemas

G.J. Klir propuso una clasificación del conocimiento de sistemas en cuatro niveles, útil para el modelado y la simulación:

- Nivel 0 - Fuente: Se identifica la porción del mundo real a modelar (sistema fuente) y las formas de observarla.
- Nivel 1 - Datos: Se dispone de una base de datos de medidas y observaciones del sistema fuente.
- Nivel 2 - Generación: Se puede recrear los datos usando una representación más compacta (fórmulas, algoritmos).
- Nivel 3 - Estructura: Se sabe cómo recrear los datos en términos de componentes interconectados (conocimiento de la estructura interna).

Los modelos formalizan el conocimiento en los Niveles 2 (generación) y 3 (estructura). Esta clasificación permite definir tres tipos de problemas relacionados con el conocimiento de sistemas: análisis, inferencia y diseño.

- Análisis: Comprender el comportamiento del sistema (existente o hipotético) usando el conocimiento de su estructura. El análisis no genera nuevo conocimiento, solo explícita lo implícito en la descripción del sistema.
- Inferencia: Conocer la estructura del sistema a partir de observaciones. Se dispone de una base de datos del comportamiento del sistema fuente. Se busca una representación del conocimiento (Nivel 2 o 3) para recrear los datos. Este proceso se denomina "construcción del modelo".
- Diseño: Investigar estructuras alternativas para un sistema nuevo o rediseñado. El sistema fuente no existe. El objetivo es construir un sistema con el comportamiento deseado. Se requiere llegar al Nivel 3 (estructura) para la construcción mediante interconexión de componentes tecnológicos.

La inferencia y diseño implican ascender en los niveles de conocimiento.

La Ingeniería Inversa combina elementos de inferencia y diseño. Se realiza un gran número de observaciones

del sistema existente. Se infiere el comportamiento del sistema y se diseña una estructura alternativa con ese comportamiento.

Marco formal para el modelado y la simulación

Se presenta un marco formal para el modelado matemático y la simulación, donde se definen cinco entidades: sistema fuente, base de datos del comportamiento, modelo, simulador y marco experimental) y dos relaciones: entre ellas, relación de modelado y relación de simulación. En la Figura 1.2 se han representan esquemáticamente las cinco entidades y las dos relaciones.

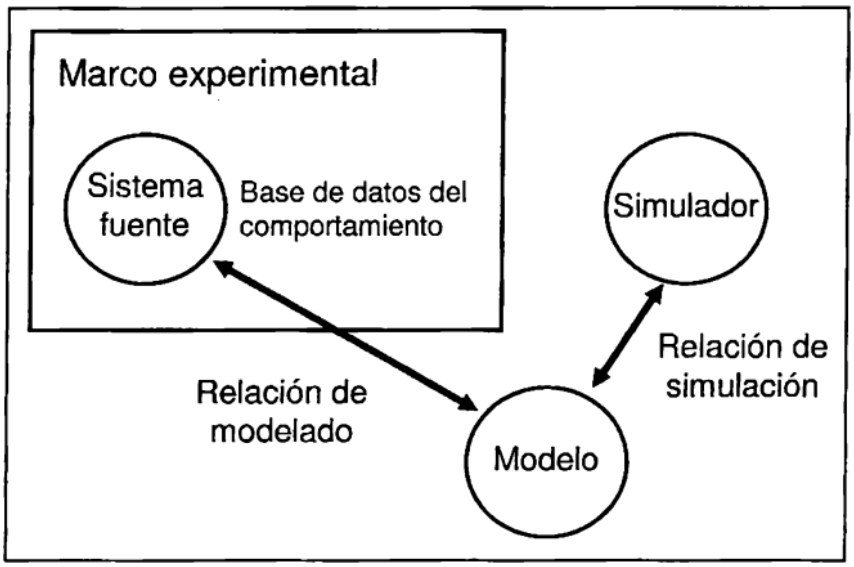

Figura 1.2: Entidades básicas del modelado y simulación, y su relación.

Figure 2: Imagen

El **Sistema Fuente** es el entorno real o virtual que se desea modelar. Representa la fuente de datos observables, en forma de trayectorias (observaciones en el tiempo) de variables. Se ubica en el Nivel 0 (fuente) de conocimiento del sistema.

La **Base de Datos del Comportamiento del Sistema** contiene los datos recopilados a partir de observaciones o experimentos con el sistema fuente. Estas observaciones corresponden al Nivel 1 (datos) de conocimiento del sistema. Los datos se obtienen bajo marcos experimentales específicos.

El **Marco Experimental** define las condiciones bajo las cuales se observa o experimenta con el sistema. Es la formulación operacional de los objetivos del proyecto de modelado y simulación.

Puede interpretarse de dos maneras:

- Definición del tipo de datos: El marco experimental especifica el tipo de datos que se incluirán en la base de datos del comportamiento del sistema.
- Sistema de medida y observación: El marco experimental es un sistema que interactúa con el sistema de interés para obtener observaciones bajo condiciones experimentales controladas. Este sistema típicamente incluye: - Generador: Produce secuencias de entrada para el sistema. - Receptor: Monitorea el experimento y verifica que se cumplan las condiciones experimentales. - Transductor: Observa y analiza las secuencias de salida del sistema.

El sistema fuente proporciona la base para la construcción de la base de datos del comportamiento del sistema. La base de datos del comportamiento del sistema, a su vez, sirve como insumo para el desarrollo del modelo y la simulación. El marco experimental define el contexto en el que se recopilan los datos y se realiza la simulación.

La comprensión de estos tres conceptos es fundamental para establecer una base sólida en el modelado y la simulación. El sistema fuente, la base de datos del comportamiento del sistema y el marco experimental proporcionan un marco conceptual para el diseño y la ejecución de estudios de modelado y simulación efectivos.

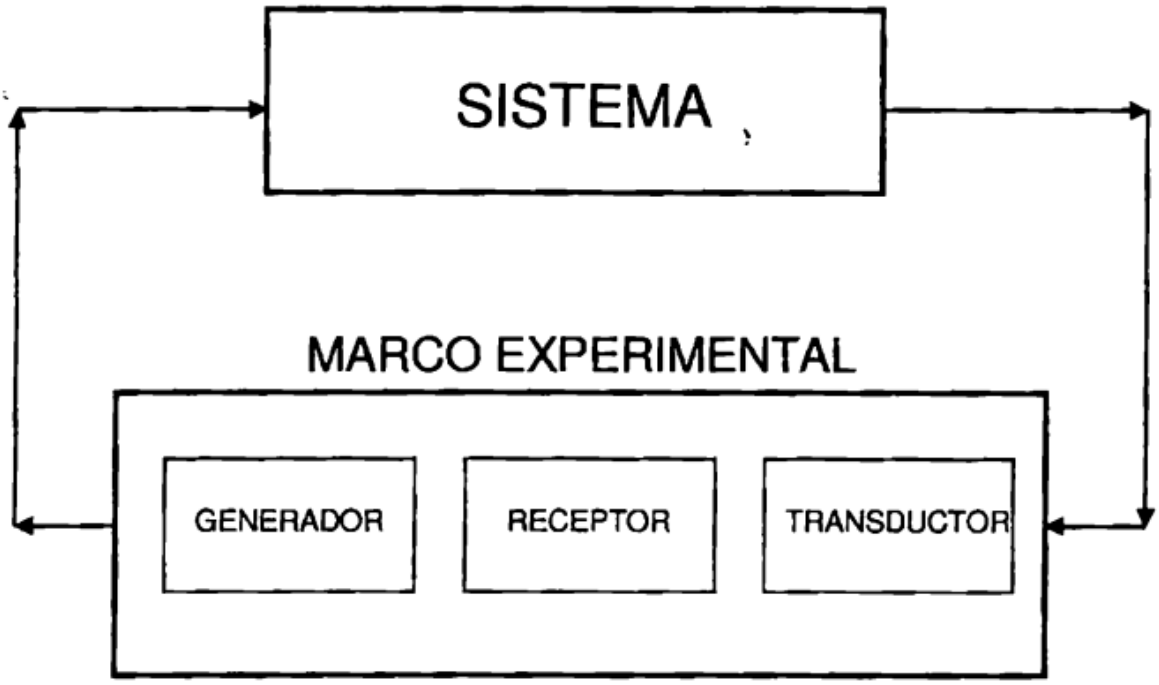

Figura 1.3: Marco experimental y sus componentes.

Figure 3: Imagen

Es fundamental establecer los objetivos del estudio al inicio del proceso de desarrollo del modelo.Los objetivos permiten enfocar el modelo en los aspectos relevantes del sistema. Definen los marcos experimentales adecuados para recopilar datos.

La elección del nivel de abstracción del modelo depende de las preguntas que se
buscan responder. Cuanto más exigentes sean las preguntas, mayor resolución se requiere en la descripción del modelo. Un nivel de abstracción apropiado es crucial para alcanzar los objetivos del estudio.

En el caso del diseño de sistemas, para transformación de objetivos a marcos experimentales:

1. Establecer las medidas de salida: Cuantifican la eficacia del sistema en cumplir sus objetivos.
2. Incluir variables de salida en el modelo: Su valor se calcula durante la simulación.
3. Cálculo de las medidas de salida: Se realiza en el transductor del marco experimental.

Las entidades del Marco Formal son:

- Modelo: Conjunto de instrucciones, reglas, ecuaciones o ligaduras para reproducir el comportamiento del sistema (Nivel 2 o 3 de conocimiento).
- Simulador:** Agente computacional capaz de ejecutar el modelo y generar su comportamiento.
- Marco Experimental:** Define las condiciones bajo las cuales se observa o experimenta con el sistema.

Las relaciones entre las Entidades son:

- Relación de Modelado (Validez): Grado en el que el modelo representa fielmente al sistema en el marco experimental de interés.
  - Validez Replicativa: El comportamiento del modelo y del sistema se corresponden dentro de una tolerancia aceptable.
  - Validez Predictiva: El modelo predice el comportamiento del sistema en situaciones no observadas.
  - Validez Estructural: El modelo reproduce el comportamiento componente a componente del sistema.
- Relación de Simulación (Corrección del Simulador): El simulador ejecuta sin errores las instrucciones del modelo.

Complejidad del Modelo:

- Recursos requeridos por el simulador para interpretar el modelo correctamente.
- Se relaciona con las propiedades intrínsecas del modelo.
- El modelado exitoso implica la simplificación válida del modelo.

Simplificación del Modelo:

- Se reduce la complejidad del modelo para facilitar su ejecución en el simulador.
- Se utilizan dos modelos: base (más capaz) y simplificado.
- Ambos modelos deben ser válidos dentro del marco experimental de interés.

Clasificaciones de los modelos matemáticos

En esta sección de muestran varias clasificaciones de los modelos matemáticos. La elección del modelo matemático más adecuado no depende del sistema en sí, ni del nivel de conocimiento que se tenga sobre él, sino de las preguntas que se pretenden responder mediante la experimentación con el modelo.

Modelo determinista o estocástico

- Modelos Deterministas:
  - Definición: Las variables de entrada son conocidas en cada instante. No existe incertidumbre en el resultado.
  - Ejemplo: Un servicio con citas preestablecidas y tiempos de servicio fijos.
- Modelos Estocásticos:
  - Definición: Al menos una variable de entrada es aleatoria. Las variables del modelo calculadas a partir de variables aleatorias también son aleatorias.
  - Ejemplo: Un parking con entradas y salidas de coches en instantes aleatorios.
  - Simulación:
    - Equivalencia con Experimentos Reales: Una réplica de la simulación es equivalente a un experimento real.
    - Variabilidad de los Resultados: Múltiples réplicas (con diferentes secuencias aleatorias) producen resultados diferentes.
    - Análisis Estadístico: El diseño y análisis de experimentos de simulación estocásticos deben considerar la incertidumbre en los resultados, utilizando técnicas estadísticas.

La comprensión de la diferencia entre modelos deterministas y estocásticos es crucial para seleccionar el modelo adecuado para un estudio de simulación y para interpretar correctamente los resultados obtenidos. Los modelos estocásticos, al incorporar la aleatoriedad del mundo real, permiten realizar estudios más realistas y completos, pero requieren un análisis estadístico cuidadoso para extraer conclusiones válidas.

Modelo estático o dinámico

- Modelos Estáticos:
  - Definición: El tiempo no juega ningún papel.
  - Tipos:
    - Deterministas: Describen el comportamiento de sistemas en estado estacionario.
    - Estocásticos: Utilizan simulaciones de Monte Carlo con números aleatorios para resolver problemas estáticos.
- Modelos Dinámicos:
  - Definición: El tiempo es una variable fundamental.
  - Tipos:
    - Tiempo Discreto: Las variables cambian en instantes específicos y permanecen constantes el resto del tiempo. Los eventos (cambios de variables) son finitos en un intervalo de tiempo finito. Los eventos ocurren en instantes predefinidos y equiespaciados.
    - Eventos Discretos: Similar al tiempo discreto, pero los eventos no necesariamente ocurren en instantes equiespaciados.
    - Tiempo Continuo: Las variables pueden cambiar de forma continua a lo largo del tiempo. Se aproximan mediante discretización temporal, transformando el modelo en uno de tiempo discreto.
- Discretización Temporal:
  - Definición: Proceso de transformar un modelo de tiempo continuo en uno de tiempo discreto.
  - Motivo: Imposibilidad de calcular el valor de las variables en infinitos instantes de tiempo.
  - Aplicación: Algoritmos calculan el valor de las variables solo en instantes específicos.

La elección entre un modelo estático o dinámico depende de la naturaleza del sistema y de las preguntas que se pretenden responder. Los modelos estáticos son útiles para analizar el comportamiento en estado estacionario, mientras que los modelos dinámicos permiten estudiar el comportamiento del sistema a lo largo del tiempo. La discretización temporal es una técnica crucial para simular modelos de tiempo continuo en computadoras.

Ejemplo: “Modelo de Tiempo Continuo”, describe que la variación en el tiempo de la variable x es una función f del valor de la propia variable, de una variable de entrada u y del valor del tiempo t.

$$\frac{dx}{dt} = f(x, u, t)$$

Figure 4: 1.1

Aplicando el método de integración explícito de Euler con un intervalo de discretización temporal delta t, se obtiene el modelo de tiempo discreto.

donde xi representa el valor de x en el instante ti, y xi-1 y ui-1 representan el valor de x y u en el instante ti-1, respectivamente. Despejando xi se obtiene:

$$\frac{x_i - x_{i-1}}{\Delta t} = f\left(x_{i-1}, u_{i-1}, t_{i-1}\right) \qquad \text{con } i = 1, 2, \ldots$$

Figure 5: 1.2

$$x_i = x_{i-1} + \Delta t \cdot f\left(x_{i-1}, u_{i-1}, t_{i-1}\right) \qquad \text{con } i = 1, 2, \ldots$$

Figure 6: 1.3

Esta expresión permite calcular el valor de la variable X en un instante de tiempo ti, conocido el valor de dicha variable y de la variable de entrada en un instante anterior ti-1. Es decir, permite calcular la evolución en el tiempo de x. La simulación del modelo consiste en el cálculo del valor de la variable x en los instantes:

$$t_i = t_0 + i \cdot \Delta t \qquad \text{con } i = 1, 2, \ldots$$

donde el valor de la variable x en el instante ti, que se representa xi,se calcula de la Ec. (1.3). Obsérvese que para ello, es preciso conocer: — El valor de la variable x en el instante to. Dicho valor, que se denomina valor inicial de la variable, se ha representado xo. — El valor de la variable de entrada en los instantes de tiempo to, tl, etc. Estos valores se han representado uo, ul, etc. Como resultado de la simulación del modelo, se obtienen los pares de valores (ti, xi) para i = 0, 1, 2, . . . , que describen la evolución en el tiempo de la variable x. Como la variable x es de tiempo continuo, su evolución suele graficarse interpolando linealmente entre los valores calculados.

Modelado y Simulación de Tiempo Discreto

Los modelos de tiempo discreto son el tipo de modelo más fácil de entender de manera intuitiva, ya que sus variables van cambiando de valor únicamente en instantes de tiempo equiespaciados. Asi, para simular estos modelos el reloj de la simulación (que indica el valor del tiempo simulado) avanza saltando un cierto intervalo de tiempo denominado paso de avance en el tiempo, que es constante a lo largo de la simulación. Si el instante inicial de la simulación es to y el paso de avance en el tiempo es delta t, entonces el reloj de la simulación va saltando sucesivamente a los instantes to, t1, t2, . . . , donde

$$t_{i+1} = t_i + \Delta t \qquad \text{para } i = 0, 1, 2, \ldots$$

En cada uno de esos instantes, el modelo se encuentra en un estado,recibe unas entradas y genera unas salidas. El modelo permite calcular, a partir de su estado actual y de sus entradas actuales, cuáles son sus salidas actuales y cuál será su estado en el siguiente instante de tiempo.

Una aplicación importante de este modelo es la descripción de circuitos digitales síncronos, en los cuales el periodo del reloj del circuito define el paso de avance en el tiempo. También se emplean como aproximaciones de modelos de tiempo continuo. En este caso, el paso de avance en el tiempo viene determinado por el intervalo de discretización temporal.

Descripción de modelos de tiempo discreto

Cuando el modelo tiene un número finito de estados y sus entradas pueden tomar un número finito de posibles valores, una forma sencilla de especificar el comportamiento del modelo es mediante su tabla de transición/salidas. Se escriben todas las posibles combinaciones de valores de los estados y las entradas, y para cada una de éstas se indica el valor de la salida y del siguiente estado. Se usa mucho para describir el comportamiento de los circuitos digitales.

Ejemplo: Consideremos el siguiente modelo de un biestable de tipo D. El modelo tiene dos estados: 0 y 1, una entrada y una salida. La entrada puede tomar dos posibles valores, representados mediante 0 y 1. Hay cuatro combinaciones entre los estados y los posibles valores de la entrada. Estas cuatro combinaciones se escriben en las dos primeras columnas de la tabla. El número de combinaciones determina el número de filas de la tabla. Para cada una de estas combinaciones, se escribe en la tercera columna el valor del estado siguiente. En la cuarta columna se escribe el valor de la salida actual del sistema.

Estado actual	Entrada actual	Estado siguiente	Salida actual
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	1

La primera fila de la tabla indica que cuando el estado actual es 0 y la entrada es 0, entonces la salida del sistema es 0 y el estado en el siguiente instante de tiempo es 0. La segunda fila indica que cuando el estado actual es 0 y la entrada actual es 1, entonces la salida actual es 0 y el estado siguiente es 1, y así sucesivamente. Obsérvese que en el biestable D la salida es igual al estado actual y el estado siguiente es igual a la entrada actual. Conocido el estado inicial del modelo, la tabla de transición/salidas permite calcular las trayectorias de estado y de salida correspondientes a una trayectoria de entrada. A continuación se muestra un ejemplo del funcionamiento del biestable D. Se comprueba que la salida sigue a la entrada, pero retrasada un paso en el tiempo.

Tiempo	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	
Trayectoria de entrada	1	0	1	0	1	0	1	0	1	0	
Trayectoria de estado	0	1	0	1	0	1	0	1	0	1	0
Trayectoria de salida	0	1	0	1	0	1	0	1	0	1	

Figure 7: Im2

Esta tabla es útil para representar el comportamiento de modelos sencillos,con un número pequeño de estados

y unas entradas con un número pequeño de posibles valores. Una forma más general de representar el comportamiento de un modelo de tiempo discreto es indicando su función de transición de estado y su función de salida. Sean xi la entrada, qi el estado e yi la salida del modelo en el instante ti, donde el indice temporal i toma los valores 0, 1, 2, ...

- Función de transición de estado, delta: describe cómo se calcula el estado siguiente a partir del estado y la entrada actuales.

$$q_{i+1} = \delta(q_i, x_i) \qquad \text{para } i : 0, 1, 2, \ldots$$

Figure 8: 1.6

- Función de salida, lambda: describe cómo se calcula la salida actual a partir del estado y la entrada actuales.

$$y_i = \lambda(q_i, x_i) \qquad \text{para } i : 0, 1, 2, \ldots$$

Figure 9: 1.7

Estas funciones permiten calcular la trayectoria del estado (ql, q2, ...) y la trayectoria de la salida (yo, yl, y2, ...), conocidos el estado inicial del modelo (qo) y la trayectoria de entrada (xo,x1, x2, . . .).

La información representada mediante una tabla de transición/ salidas puede ser descrita de forma más compacta empleando las funciones de transición de estado y de salida. Por ejemplo, el comportamiento del biestable D (la salida es igual al estado actual y el estado siguiente es igual a la entrada actual) puede representarse de la forma siguiente:

$$\delta(q, x) = x$$

$$\lambda(q, x) = q$$

Figure 10: 1.8.9

Simulación de modelos de tiempo discreto

El siguiente algoritmo es un simulador para un modelo de tiempo discreto descrito mediante las funciones de transición de estado y de salida. El algoritmo calcula las trayectorias del estado y de salida del modelo, a partir de la trayectoria de entrada y del estado inicial.

```
iFin = 9                      índice temporal final
x(0) = 1, ..., x(9) = 0       tráyectoria de entrada
q(0) = 0                      estado inicial
i = 0                         inicialización del índice temporal
while ( i <= iFin ) {
   y(i)   = λ( q(i), x(i) )
   q(i+1) = δ( q(i), x(i) )
   i      = i + 1
}
```

Figure 11: Im4

Autómatas celulares

El autómata celular es un tipo de modelo con dos propiedades características: - Posee una estructura regular: está compuesto de componentes iguales, conectados de acuerdo a un cierto patrón espacial. - El comportamiento de todos los componentes está regido por el mismo conjunto de reglas.

Fue empleado por von Neumann y Ulam para describir la autorreproducción de sistemas biológicos. Cada uno de los componentes iguales que componen el autómata celular se denomina "célula" y el conjunto de células "espacio celular". La distribución espacial de las células puede ser un mallado unidimensional, bidimensional o multidimensional, conectado de manera uniforme. Las células que influyen sobre una célula en particular, denominadas sus vecinas, son a menudo aquellas situadas más cerca en el sentido geométrico.

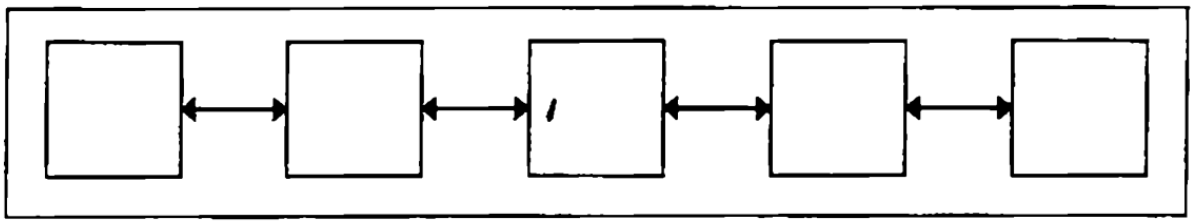

Figura 1.4: Espacio celular unidimensional.

Figure 12: Im5

Un autómata unidimensional se puede hacer conectando las células en fila, de modo que cada célula tenga conectada una célula a su izquierda y otra a su derecha. Supongamos que cada célula de este autómata unidimensional puede estar en dos estados, 0 y 1, y que recibe como entrada los estados de las células vecinas. Hay 8 posibles combinaciones de los valores de las 2 entradas y del estado de la célula. La tabla de transición

de estados de la célula, qué describe el nuevo estado de la célula en función de sus entradas y del estado actual, tendrá la forma mostrada a continuación.

Entrada izquierda actual	Estado actual	Entrada derecha actual	Estado siguiente
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

Figure 13: Im6

La cuarta columna contiene interrogaciones: dicha columna tiene 8 filas y la célula puede estar en 2 estados, hay 2ˆ8 = 256 posibles tablas de transición de estados. Cada una de estas posibles tablas representa una regla. Criterio para designar las reglas: la cuarta columna puede interpretarse como un número binario de 8 bits, de modo que la primera fila sea el bit menos significativo y la octava fila el más significativo. El nombre de la regla se obtiene convirtiendo este número de decimal. Para simular el autómata hay que asignar valor inicial al estado de cada célula del espacio celular y aplicar el algoritmo simulador, que calcula en cada instante el estado de cada célula a partir del estado en el instante anterior de esa célula y de sus vecinas. Por ejemplo, en la Figura 1.5 se muestra la tabla de transición de estados del autómata con regla 90 y un ejemplo de su simulación. Para facilitar la visualizacióndel resultado de la simulación, el estado 1 se ha representado mediante un aspa y el estado 0 mediante un guión.

Entrada izquierda actual	Estado actual	Entrada derecha actual	Estado siguiente
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

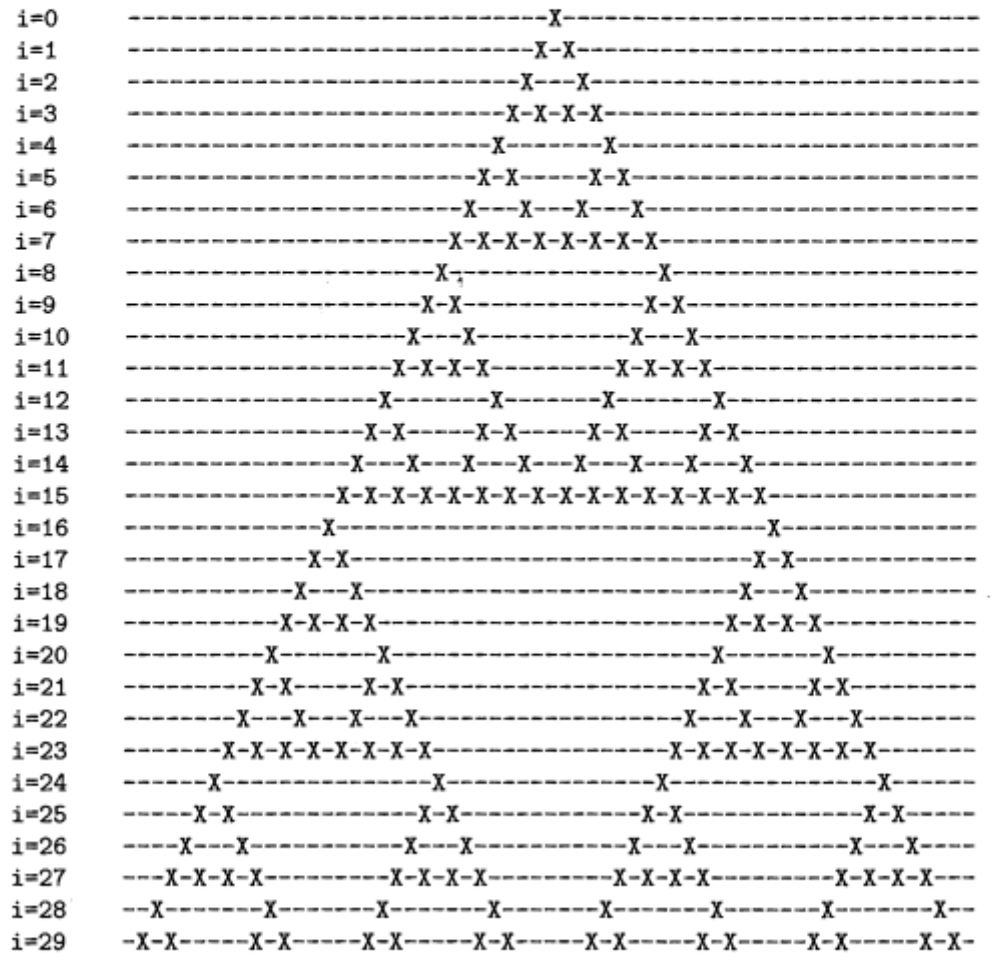

Figura 1.5: Autómata celular con regla 90: tabla de transición de estados (arriba) y simulación de un espacio celular con esa regla compuesto por 61 células (abajo).

Figure 14: Im8

Los autómatas celulares unidimensionales exhiben una variedad de comportamientos. Se distinguen principalmente cuatro tipos: extinción rápida de la dinámica, comportamiento periódico, comportamiento caótico y patrones regulares interesantes pero no predecibles.

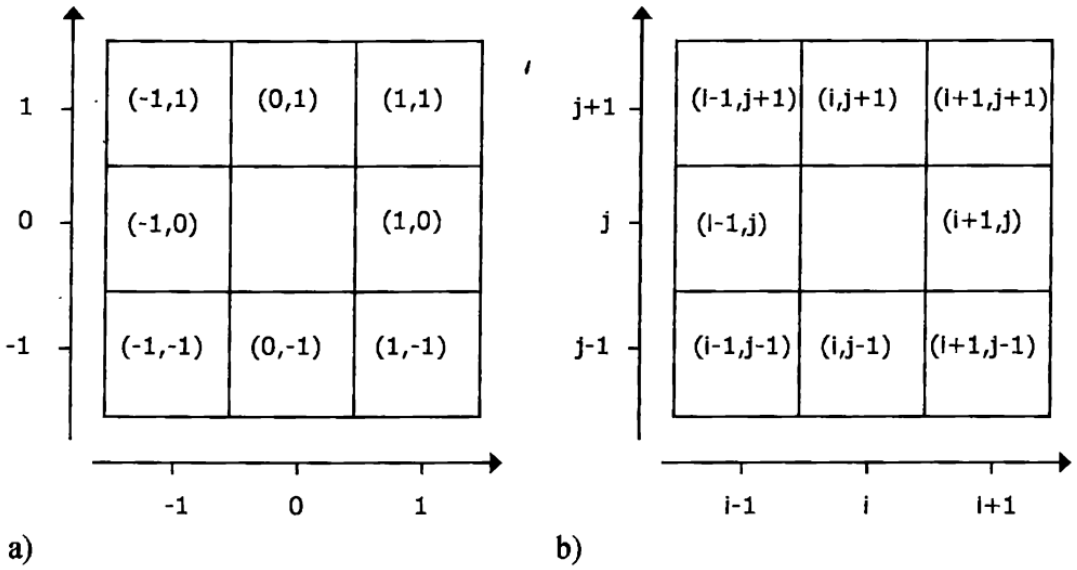

Figura 1.6: Células vecinas en el *Juego de la Vida* a la situada en: a) $(0,0)$; y b) (i,j).

Figure 15: Im7

Ejemplo, de autómata bidimensional: "Juego de la Vida de Conway", cuyo tamaño puede ser finito o infinito. Cada celda está acoplada a aquellas que se encuentran más próximas a ella, tanto lateral como diagonalmente. Para una célula situada en el punto (0,0), sus células vecinas laterales están situadas en los puntos (0,1), (1,0), (0, —1)y (—1,0), y sus células vecinas diagonales están situadas en (1,1), (—1,1), (1, —1) y (—1,—1), como en la Figura 1.6a. En 1.6b se muestran las células vecinas de una célula situada en la posición (i,j). El estado de cada célula puede tomar dos valores: 1 (Viva)y 0 (muerta). Cada una de las células puede sobrevivir (está viva y permanece viva), nacer (su estado pasa de 0 a 1) y morir (su estado pasa de 1 a 0) a medida que el juego progresa.

Las reglas son las siguientes:

- Una célula permanece viva si tiene en su vecindad 2 ó 3 células vivas.
- Una célula muere debido a superpoblación si hay más de 3 células vivas en su vecindad.
- Una célula muere a causa del aislamiento si hay menos de 2 células vivas en su vecindad.
- Una célula muerta vuelve a la vida si hay exactamente 3 células vivas en su vecindad.

El espacio celular tiene dimensiones finitas, como un cuadrado de N x N células. Es necesario definir el comportamiento de las células en los bordes, ya que no tienen todos sus vecinos. Una opción es mantener un estado constante en los bordes, como todas las células muertas. Otra alternativa es aplicar un enfoque toroidal, donde el índice N se interpreta también como 0.

El modelo del Juego de la Vida se desarrolla en un tiempo discreto, avanzando en pasos sucesivos. Comienza con una configuración inicial de células vivas y, a medida que avanza la simulación, la disposición de las células cambia. El objetivo del juego es descubrir nuevos patrones y analizar su comportamiento, ilustrado en la Figura 1.7.

El algoritmo para simular el autómata consiste en almacenar el estado inicial de las células en una estructura de datos. En cada paso de tiempo, se examinan todas las células y se aplica la función de transición de

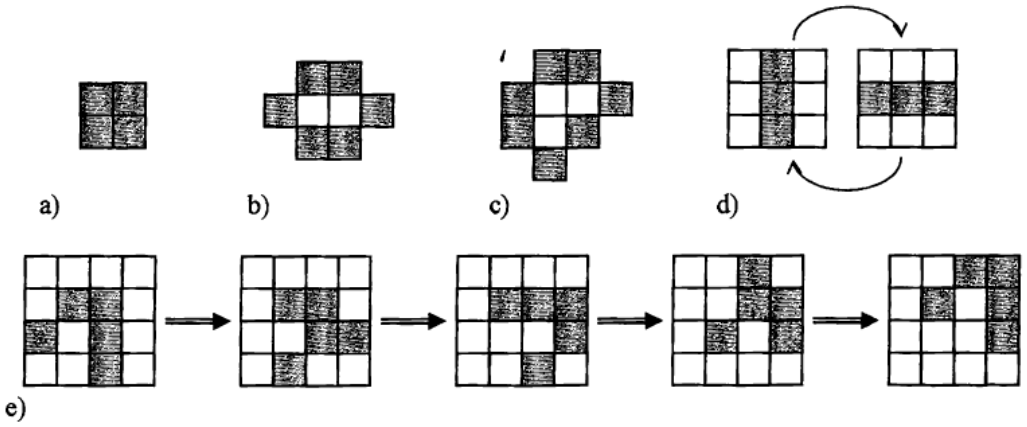

Figura 1.7: Algunos patrones que aparecen en el *Juego de la Vida de Conway*. Los patrones a), b) y c) son estables, ya que no cambian. El patrón d) es oscilante. En e) se muestra un ciclo de patrones que se mueve.

Figure 16: Im9

estado, guardando el siguiente estado en una segunda copia de la estructura de datos. Luego, este siguiente estado se convierte en el estado actual y se avanza un paso en la simulación.

El algoritmo original calcula el nuevo estado de todas las células en cada paso de tiempo. Una optimización posible es calcular solo el nuevo estado de las células cuyo estado puede cambiar. Esto implica mantener un registro de qué células pueden cambiar en el siguiente paso, basado en si el estado de sus células vecinas cambió en el paso actual.

La versión optimizada del algoritmo identifica las células cuyo estado cambia en una transición y luego determina el conjunto de todas las células vecinas de estas células. Este conjunto incluye todas las células que podrían cambiar su estado en el próximo paso. Para determinar si cada una de estas células realmente cambia de estado, se evalúa su función de transición de estado.

Los autómatas celulares se utilizan para modelar sistemas cuyo comportamiento resulta de la interacción entre sus partes. Tienen diversas aplicaciones, como el estudio del crecimiento de cristales, la propagación de incendios forestales y manchas en el mar, reacciones químicas, transporte en medios fluidos y sólidos, tráfico vehicular urbano, comportamiento de colonias de seres vivos, y el crecimiento de corales y conchas, entre otros ejemplos.

Modelado y simulación de eventos discretos

Tanto los modelos de tiempo discreto como los de eventos discretos comparten la característica de que las variables del modelo cambian solo en momentos específicos, manteniéndose constantes el resto del tiempo. La diferencia radica en que en los modelos de tiempo discreto, estos momentos están equiespaciados en el eje temporal, mientras que en los modelos de eventos discretos no existe esta restricción. En estos últimos, el tiempo entre eventos consecutivos puede variar, por lo que se necesita un calendario de eventos para registrar los momentos de activación de los eventos planificados.

En la sección se presentan dos metodologías para describir modelos de eventos discretos: el modelado orientado a la planificación de eventos y el modelado orientado a los procesos.

Modelado orientado a la planificación de eventos

Los modelos de eventos discretos se definen especificando qué eventos pueden ocurrir y cómo están relacionados entre sí. Un evento puede cambiar el estado del modelo, programar eventos futuros y cancelar la programación de eventos.

A continuación se describe una manera sencilla de programar, empleando un lenguaje de programación imperativo. El programa podría constar de las siguientes reglas:

- Rutina de inicialización. Se asignan valores iniciales a las variables de estado, se inicializan los acumuladores estadísticos y se planifican eventos añadiéndolos al calendario de eventos. Los acumuladores estadísticos son variables intermedias utilizadas para calcular las variables de salida de la simulación, como la suma de tiempos de espera en una cola o el número de entidades procesadas por un recurso.
- Rutina de tiempo. Determina cuál es el evento más inminente de los planificados en el calendario de eventos y avanza el reloj de la simulación hasta ese instante.
- Rutina de eventos. Son las rutinas, una para cada tipo de evento, que realizan el flujo de acciones asociado al evento. Puede modificar el valor de las variables y acumuladores estadísticos, así como añadir o quitar eventos del calendario de evento.
- Rutina de informes. Al finalizar la simulación, calcula y muestra el valor de las variables de salida.

El programa principal controla el flujo de control de la simulación, tal como se muestra en la Figura 1.8.

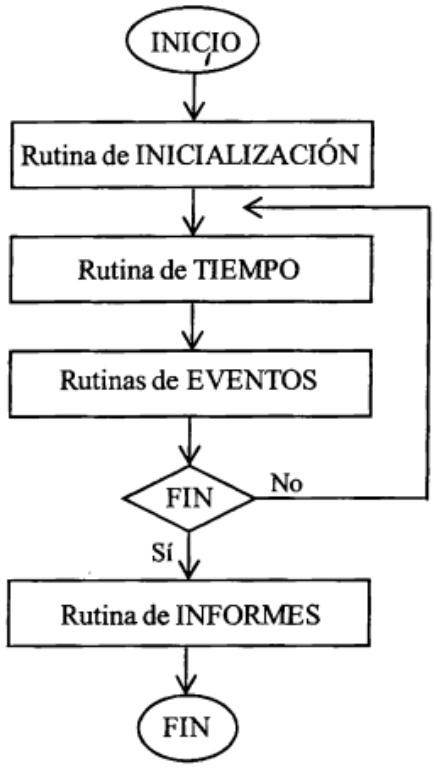

Figura 1.8: Flujo de la simulación de un modelo orientado a la planificación de eventos.

Figure 17: Im10

1. Comienza la simulación.

2. La rutina de inicialización del programa principal inicializa el reloj de la simulación, las variables de estado, el calendario de eventos y los acumuladores estadísticos. Se activa el evento "Inicio de la Simulación", que planifica otros eventos para ejecución en el futuro, agregándolos al calendario de eventos en orden cronológico.
3. Tras la inicialización, el programa pasa a la rutina de tiempo. Avanza el reloj de la simulación hasta el momento del próximo evento en el calendario, el cual se elimina y activa.
4. El programa principal pasa el control a la rutina asociada al tipo de evento activado. Esta rutina actualiza variables de estado, acumuladores estadísticos y añade o elimina eventos en el calendario. El evento "Finalización de la Simulación" se activa cuando se cumplen condiciones para terminar la simulación, como alcanzar un tiempo específico o ciertas condiciones del sistema. Una acción importante en este evento es calcular las variables de salida basadas en los acumuladores estadísticos.
5. Si se activa el evento "Finalización de la Simulación", el programa principal pasa a la rutina generadora de informes. De lo contrario, retorna el control a la rutina de tiempo.
6. La rutina generadora de informes muestra el valor de las variables de salida de la simulación.
7. Finaliza la simulación.

Esta manera de describir y programar modelos de eventos discretos es simple conceptualmente para modelos simples. Sin embargo, se vuelve complicada y propensa a errores al aplicarla a modelos grandes, donde se deben definir muchos tipos de eventos y sus relaciones causales.

Se han desarrollado formalismos orientados a la planificación de eventos para simplificar la descripción y simulación de modelos de gran escala. Estos formalismos permiten describir los modelos de manera modular y jerárquica, como componentes interactivos, para reproducir la estructura del sistema.

En lugar de describir todo el sistema de manera global, se divide en partes más simples. Cada parte se modela independientemente y luego se describe cómo interactúan. Este enfoque modular y jerárquico facilita el desarrollo y prueba de modelos por separado, permitiendo su reutilización en diferentes contextos de aplicación.

Para describir modelos de forma modular y jerárquica, es necesario definir los componentes del modelo y cómo interactúan entre sí. Esta interacción puede ser especificada mediante dos tipos de acoplamientos.

- Acoplamiento modular. Cada componente tiene puertos de entrada y salida que forman su interfaz para la interacción con otros componentes. La conexión entre los puertos de salida y entrada facilita la transmisión instantánea de información. Los modelos compuestos, formados por componentes interconectados, también tienen interfaces con puertos de entrada y salida, donde la transmisión de información es igualmente instantánea.
- Acoplamiento no modular. La interacción entre componentes se describe por la influencia del estado de unos (influyentes) en la transición del estado de otros (influenciados). Los componentes con acoplamiento no modular carecen de interfaz, ya que el estado de los influyentes incide directamente en las funciones de transición de los influenciados. Un ejemplo son las células en autómatas celulares, que están acopladas de forma no modular.

Modelado orientado a los procesos

El modelado orientado a los procesos busca simplificar la descripción de modelos, acercándola al razonamiento humano. Se enfoca en las entidades y su movimiento a través del sistema, centrándose en los procesos que realizan estas entidades.

La metodología de orientación a los procesos se popularizó en la década de 1970 con la llegada de lenguajes de simulación general para modelos de eventos discretos. Estos lenguajes permiten traducir automáticamente descripciones orientadas a procesos a una planificación de eventos escrita en lenguajes de programación. Aunque el código ejecutable siempre se orienta a la planificación de eventos, algunos de estos lenguajes, como GPSS, SIMSCRIPT, SLAM, SIMAN, siguen siendo utilizados hoy en día.

Actualmente, el modelado orientado a procesos se realiza típicamente utilizando entornos de simulación, que son software construido sobre lenguajes de simulación para facilitar la descripción del modelo con interfaces intuitivas. Ejemplos de estos entornos incluyen AnyLogic, Arena, AutoMod, Enterprise Dynamics, ExtendSim, FlexSim, ProModel y SIMUL8.

Estos entornos posibilitan al usuario construir modelos instanciando módulos predefinidos y conectándolos gráficamente. Permiten visualizar la evolución del modelo durante la simulación mediante animación y capacidades gráficas. Además, la interfaz gráfica de usuario facilita acceder a niveles inferiores en la descripción del modelo, incluyendo la descripción de partes del modelo utilizando el lenguaje de simulación e incluso el lenguaje de programación.

El entorno de simulación Arena permite utilizar diferentes niveles de descripción: elementos de alto nivel parametrizables por el usuario y elementos de bajo nivel que pueden definirse utilizando el lenguaje de simulación SIMAN, así como los lenguajes de programación Visual Basic y C/C++.

Algunos entornos ofrecen herramientas para el modelado estadístico de entradas aleatorias, la definición de experimentos y problemas de optimización, así como el análisis estadístico de los resultados.

Modelado de una oficina de atención al público

El modelo a seguir muestra las distinciones entre el modelado orientado a eventos y el modelado orientado a procesos, a través de una oficina de atención al público con un empleado. La estructura lógica del modelo es la siguiente (véase la Figura 1.9).

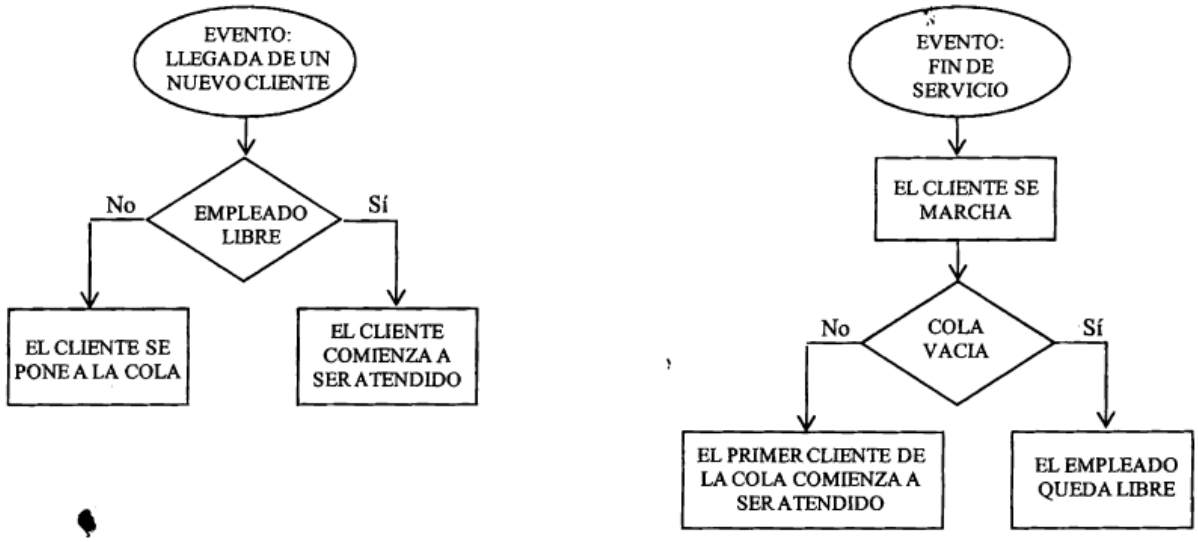

Figura 1.9: Esquema de la estructura lógica del modelo.

Figure 18: Im11

- Cuando un nuevo cliente llega y el empleado está ocupado, el cliente se une al final de la cola. Si el empleado está libre, el cliente es atendido de inmediato. La cola opera bajo la disciplina FIFO ("First In, First Out"), lo que significa que los clientes son atendidos en el orden en que llegaron a la cola.
- Cuando el empleado termina de atender a un cliente, este se va y el siguiente cliente en la cola comienza a ser atendido. Si la cola está vacía, el empleado permanece libre hasta que llegue un nuevo cliente.

La estructura del modelo incluye una cola y un proceso con un recurso: el empleado. Se representan los eventos de entrada y salida al sistema, donde los eventos de entrada (llegada de clientes) y de salida (clientes

atendidos que abandonan) están dispuestos en una trayectoria temporal. Cada evento se muestra como una flecha vertical en el eje temporal en el momento de su ocurrencia. Esto se puede ver en la Figura 1.10.

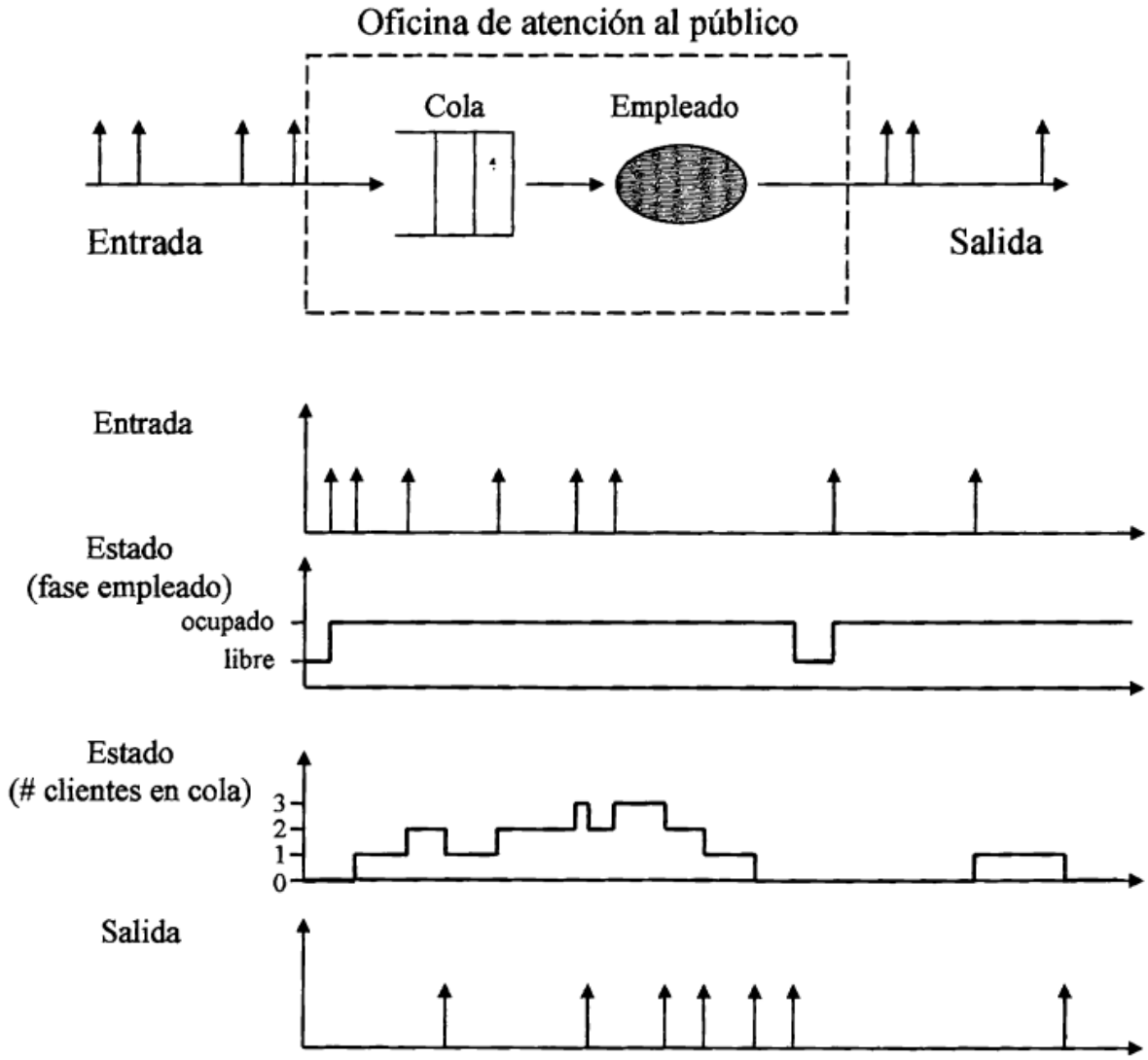

Figura 1.10: Ejemplo del comportamiento del modelo de la oficina.

Figure 19: Im12

Los procesos de llegada y atención al cliente se modelan asumiendo que llegan de manera individual y que el tiempo entre llegadas sucesivas sigue una distribución exponencial con una media de 10 minutos. Además, el tiempo de atención del empleado se modela como una variable aleatoria uniformemente distribuida entre 5 y 10 minutos.

Para describir el estado del sistema se requieren al menos dos variables de estado: la fase del empleado (libre u ocupado) y el número de clientes en cola (enteros positivos, incluyendo cero). La evolución del estado del modelo se ilustra en la Figura 1.10 a medida que los clientes llegan y abandonan la oficina.

El objetivo de la simulación es calcular el tiempo promedio de espera en la cola y el número promedio de clientes en espera. Los cálculos a realizar incluyen:

- tiempo medio de espera del cliente en la cola:

$$\hat{d}(n) = \frac{1}{n}\sum_{i=1}^{n} D_i$$

Figure 20: 1.10

donde n es el número total del clientes que han abandonado la cola y Di es el tiempo de espera en la cola del clientei. Es preciso llevar registro a lo largo de la simulación tanto del número de clientes que han abandonado la cola hasta ese momento, como de la suma de sus tiempos de espera.

- número medio de clientes en la cola:

$$\hat{q}(T) = \frac{1}{T}\int_{0}^{T} Q(\tau) \cdot d\tau$$

Figure 21: 1.11

donde T es el tiempo que ha durado la simulación y Q (tau) es el número de clientes que había en la cola en el instante tau. Si se representa el número de clientes en cola Q en función del tiempo, la integral es igual al área bajo la curva entre los instantes inicial y final de la simulación. Dicho área debe irse calculando a lo largo de la simulación, para lo cual se emplea el acumulador estadístico R, que inicialmente R vale cero. Se actualiza el valor de R cada vez que cambia el tamaño de la cola y en el instante final la simulación. Para ello, se suma al valor actual de R el producto del número de clientes en cola antes del cambio por el tiempo transcurrido desde el anterior cambio. Es preciso definir un segundo acumulador estadístico, tevento, que almacene el instante en que se produjo el cambio en el tamaño de la cola anterior al actual.

La descripción del modelo utilizando la metodología de planificación de eventos implica definir la condición de activación y las acciones correspondientes para cada tipo de evento. En este caso, hay cuatro tipos de eventos: inicio de la simulación, llegada de un nuevo cliente a la oficina, finalización del servicio a un cliente y finalización de la simulación. Los flujos de acciones asociados a los primeros tres tipos se representan en la Figura 1.11.

La fase del empleado se representa mediante la variable de estado E, que puede tomar los valores 0 (libre) y 1 (ocupado). El modelo tiene otras dos variables: Q (número de clientes en cola) y tevento (instante en que se produjo el último evento).

El calendario de eventos guarda dos valores: llegada (instante en que está planificada la llegada del próximo cliente) y marcha (instante en que está planificado que el cliente que está siendo atendido abandone el sistema).

El modelo tiene tres contadores estadísticos: D (suma de los tiempos de espera en cola de todos los clientes que la han abandonado), n (número de clientes que han comenzado a ser atendidos) y R (área bajo Q).

A continuación veremos cómo se describe el funcionamiento de la oficina usando la metodología de la orientación al proceso y el entorno de simulación Arena. Se toma el punto de vista de los clientes y se describe

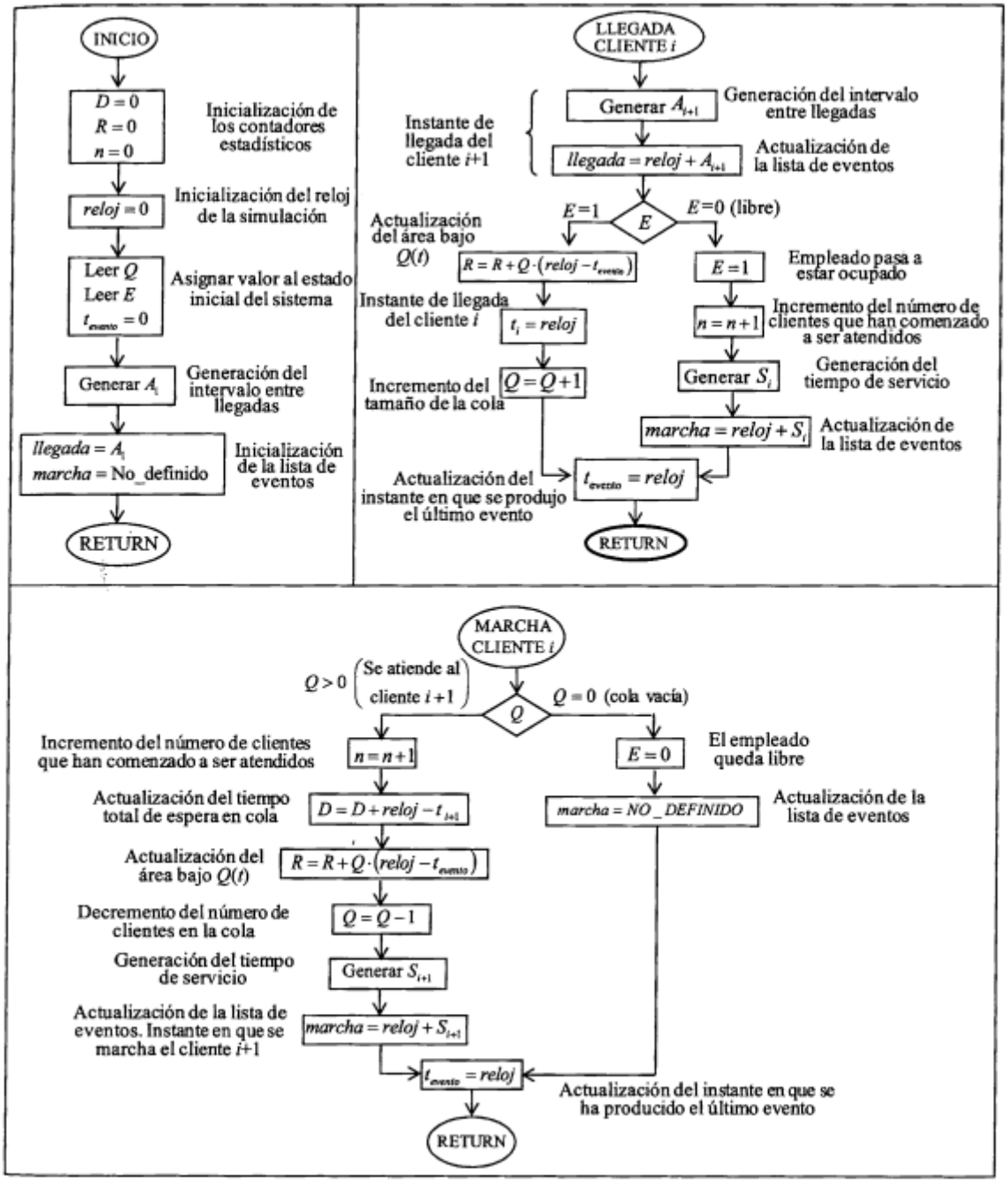

Figura 1.11: Flujos de acciones asociadas a los eventos.

Figure 22: Im13

su circulación a través del sistema. Los pasos que sigue un cliente en la oficina son los siguientes:

1. Llego a la oficina.
2. Me pongo al final de la cola. Espero hasta que yo sea el primero de la cola y el empleado esté libre.
3. El empleado me atiende durante el tiempo que requiero.
4. Finaliza mi tiempo de servicio y abandono la oficina.

El modelo se puede describir de manera simple utilizando los módulos predefinidos de Arena para representar el proceso de llegada de clientes, la atención al cliente con su cola, y la salida de clientes de la oficina. La figura 1.12a muestra el diagrama de módulos del modelo con la animación correspondiente en un momento específico de la simulación. El diagrama está compuesto por la instanciación y conexión de los tres tipos de módulos mencionados, definiendo así el flujo de clientes en la oficina. En el instante mostrado en la figura, habían llegado 109 clientes, 3 esperaban en la cola, uno estaba siendo atendido y 105 ya habían abandonado la oficina.

Al hacer doble clic en cada módulo del diagrama, se abre un menú de configuración específico. La Figura 1.12b muestra el menú de configuración del módulo de llegada de clientes (tipo "Create"), mientras que la Figura 1.12c muestra el menú del módulo que describe el proceso de atención al cliente (tipo "Process").

Arena calcula automáticamente ciertas medidas estadísticas del sistema, como los tiempos medios, máximo y mínimo en las colas, la ocupación de los recursos, entre otros. También permite al usuario definir sus propios cálculos. Al finalizar la simulación, Arena genera informes automáticos que muestran esta información.

Pasos en un estudio de simulación

Se describen los pasos que típicamente se siguen empleando modelos de eventos discretos de tipo estocástico. Los pasos son los siguientes:

1. Definición del problema: se define por qué se estudia el sistema, qué objetivos se persiguen y cuáles son las preguntas a contestar.
2. Planificación del proyecto: se estima qué recursos son necesarios para llevar a cabo el estudio: dinero, personal, hardware de computación, recursos software, etc. Si no están disponibles debe replantearse el alcance del estudio.
3. Definición del sistema: se definen qué aspectos de la realidad constituyen el sistema bajo estudio.
4. Formulación conceptual del modelo: se desarrolla un modelo preliminar, gráficamente (diagramas de bloques) o en pseudocódigo donde se definen los componentes, las variables descriptivas y las interacciones que constituyen el modelo del sistema.
5. Diseño preliminar del experimento: se define qué acción se va a ejercer sobre el modelo y cómo se va a medir su comportamiento. Se trata de definir qué variables son las entradas y cuáles las salidas, cómo va a modificarse el valor de las entradas y cómo van a recogerse los datos de salida.
6. Preparación de los datos de entrada: se establece el procedimiento a seguir para asignar valores a cada una de las variables de entrada.
7. Traducción del modelo: programando directamente el algoritmo de simulación del modelo en algún lenguaje de programación, o empleando un lenguaje o entorno de simulación.
8. Verificación y validación: se estudia si el modelo opera como debería, y si la salida del modelo es creíble y representativa del comportamiento del sistema.
9. Diseño experimental final: se diseña un experimento que proporcione la información necesaria para poder contestar a las preguntas planteadas en el estudio. El experimento consiste en cierto número de réplicas de la simulación en condiciones bien definidas.
10. Experimentación: se realiza el experimento de simulación diseñado antes.

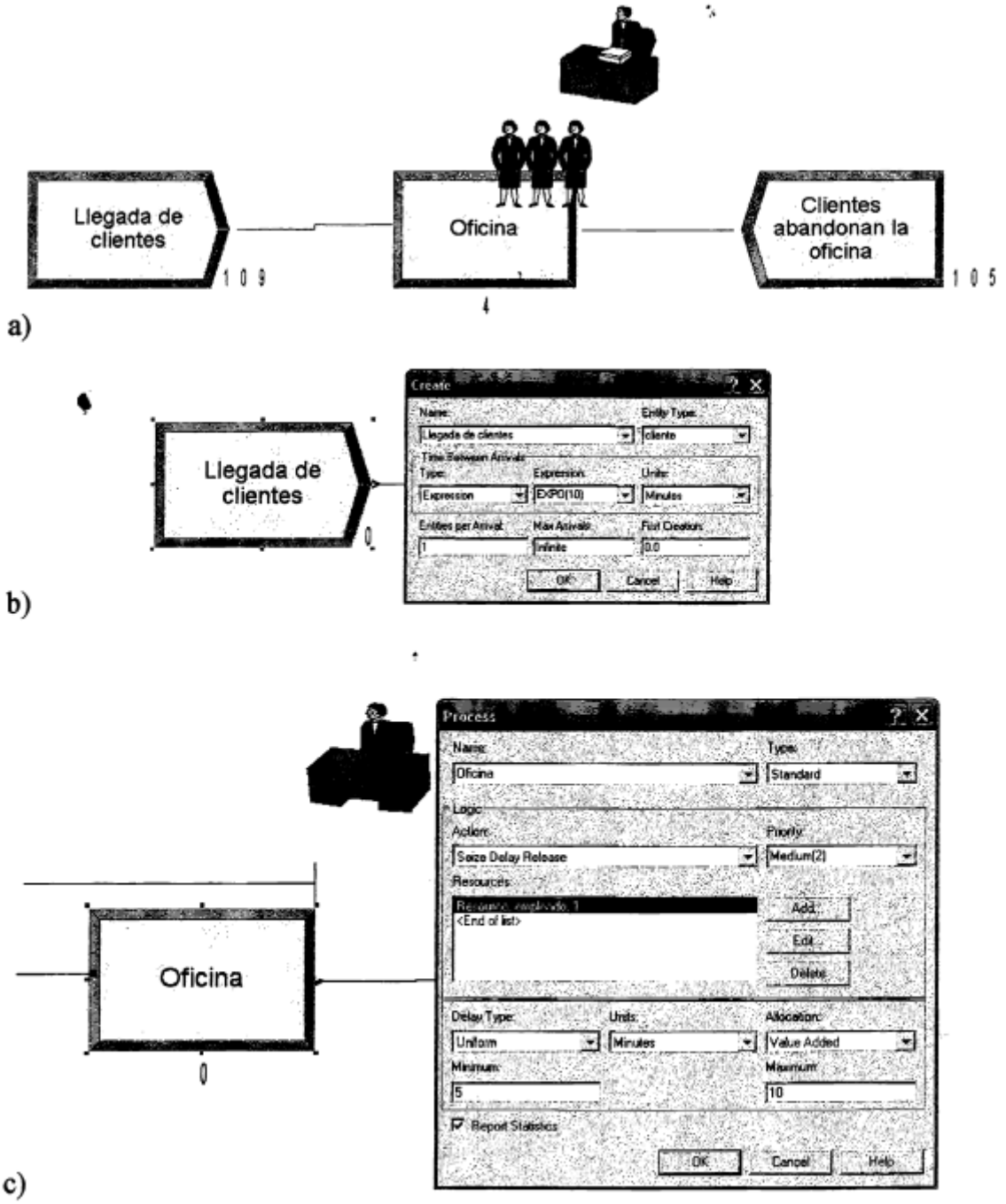

Figura 1.12: Modelado en Arena de la oficina atendida por un empleado: a) diagrama de módulos del modelo; y b) y c) propiedades de los módulos que definen los procesos de llegada de los clientes y de atención a los clientes, respectivamente.

Figure 23: Im14

11. Análisis e interpretación: se extraen conclusiones de los resultados.
12. Documentación y actuación: una vez obtenidos los resultados, analizados y extraídas las conclusiones, deben ser presentadas de la forma más adecuada. El modelo debe ser documentado y preparado de modo que pueda ser reutilizado para posteriores estudios. Finalmente, deben tomarse acciones (si era la finalidad) a partir de las conclusiones obtenidas.

La "Regla 40-20-40" proporciona una orientación acerca de cómo distribuir el esfuerzo en un estudio de simulación. Aconseja dedicar el 40 % del esfuerzo y del tiempo del proyecto en los pasos 1 a 6, el 20% del esfuerzo en el paso 7 y el 40 % restante en los pasos 8 a 12.

Definición del problema y planificación del proyecto

Formular adecuadamente un problema es crucial antes de intentar resolverlo. Este paso es esencial para comprender el objetivo del estudio de simulación y construir un modelo que pueda responder las preguntas específicas sobre el sistema real.

Ejemplos:

- Evaluación del comportamiento de determinado diseño del sistema en términos absolutos frente a determinados criterios específicos.
- Comparación del comportamiento de diferentes diseños alternativos del sistema, o comparación de diferentes políticas de operación o procedimientos.
- Predicción del comportamiento del sistema bajo ciertas condiciones.
- Análisis de la sensibilidad: determinar entre varios factores cuál es el que más afecta al comportamiento del sistema.
- Optimización: cálculo de los valores de determinados parámetros del modelo que producen que el sistema tenga la mejor respuesta.
- Estudio de las relaciones funcionales entre determinadas variables del modelo.
- Localización y análisis de los cuellos de botella: se descubre la localización de los cuellos de botella del sistema y se evaluan propuestas para descongestionar alguno de ellos.

Estos objetivos pretenden responder preguntas del tipo:

- ¿Funciona el sistema del modo que queremos que lo haga?
- ¿Por qué no funciona el sistema y qué podemos hacer para que funcione?
- ¿Cuál es la mejor alternativa?

El propósito del estudio de simulación influye significativamente en el diseño del modelo y del experimento. Si se busca evaluar el comportamiento del sistema en términos absolutos, el modelo debe reflejar con precisión el sistema real. En cambio, si se pretende comparar diferentes diseños, el modelo puede ser válido en un sentido relativo aunque no coincida absolutamente con el sistema real. Por lo tanto, el diseño del modelo, su validación, la planificación de los experimentos y el análisis de los resultados deben estar estrechamente ligados al propósito del estudio.

El objetivo del estudio de simulación requiere una definición detallada, considerando todos los aspectos relevantes del proyecto. En ocasiones, el estudio es llevado a cabo por un grupo de ingenieros que deben recomendar una solución, pero no toman la decisión final. En estos casos, los resultados deben servir para formular la recomendación y respaldarla. Es importante considerar los intereses de quien tomará la decisión final, los aspectos bajo su control, el proceso de toma de decisiones, las personas involucradas (especialmente aquellas que se oponen a cambios) y sus objetivos relevantes.

Es fundamental considerar todos los factores críticos para el éxito del proyecto durante la planificación, incluyendo:

- ¿Están claramente definidos los objetivos del estudio? ¿Por qué se realiza el estudio? ¿Qué se pretende obtener del estudio?
- ¿Se dispone de los recursos adecuados y suficientes para poder realizar el estudio: tiempo, personal, software, ordenadores, etc.?

-¿Dispone el estudio del soporte necesario por parte de la dirección? Las partes involucradas se implicarán en mayor medida si son conscientes del respaldo de la dirección. ¿Se les ha informado adecuadamente? ¿Son conscientes de que ese respaldo existe?

- ¿Se han definido las tareas de que consta el proyecto, la vinculación entre ellas, quién debe realizarlas y en qué fechas?
- ¿Posee cada uno de los miembros del proyecto la formación necesaria para realizar con éxito su cometido? ¿Está suficientemente capacitado el coordinador del proyecto?
- ¿Se han identificado los usuarios potenciales o clientes de los resultados del estudio? ¿Han sido todos ellos consultados? ¿Se ha establecido el procedimiento para comunicarles regularmente el estado en que se encuentra el proyecto y los progresos realizados?
- ¿Se han establecido los adecuados canales de comunicación entre los miembros del equipo, la dirección y los clientes?

Definición del sistema y formulación del modelo

El arte del modelado radica en la abstracción y la simplificación, buscando identificar el conjunto mínimo de características del sistema para cumplir con los objetivos del estudio. La metodología para construir el modelo podría ser la siguiente:

1. La selección de las variables de salida es relativamente simple una vez que se ha definido el objetivo del estudio.
2. La construcción del modelo implica identificar los componentes del sistema que influyen en las variables de salida y decidir si deben incluirse en el modelo o ser considerados parte del entorno, representándolos mediante entradas al modelo.
3. Para determinar las relaciones funcionales entre los componentes del modelo, es decir, la lógica del modelo, se utilizan gráficos o pseudocódigo para describir el sistema como un diagrama de flujo lógico.

Para diseñar el modelo de simulación de manera efectiva, es importante contar con la máxima información posible sobre el sistema. Esto se puede obtener de diversas fuentes, como:

- La documentación de las especificaciones de diseño del sistema, aunque a veces desactualizada e incompleta, suele ser un punto de partida razonable para comprenderlo.
- Entrevistas con operarios, ingenieros o diseñadores del sistema, si está en fase de diseño, proporcionan información valiosa. Comparar datos de salida del sistema real y del modelo, sin conocer su origen, puede ayudar a mejorar el modelo.
- La observación directa y la toma de medidas proporcionan la información más fiable. Tras consultar la documentación y discutir con expertos, es recomendable observar el sistema en funcionamiento para comprenderlo mejor.

El modelado busca un equilibrio entre la representación detallada del sistema real y la simplicidad. El modelo debe ser lo suficientemente detallado para abordar las preguntas planteadas, pero no debe incluir componentes innecesarios. El objetivo es responder a las preguntas, no replicar exactamente el sistema real, ya que agregar complejidad no siempre aumenta el realismo.

En un caso real descrito por Law y Kelton en 2000, una fábrica de comida para perros encargó un modelo de su línea de fabricación que producía alrededor de un millón de latas de comida por día. Inicialmente, cada lata de comida se representó como una entidad independiente, lo que resultó en un modelo costoso y poco

útil. Más tarde, el modelo se reescribió tratando el proceso de fabricación como un flujo continuo, lo que produjo resultados precisos y redujo significativamente el tiempo de ejecución.

En 1985, se intentó modelar una fábrica de gran tamaño que tenía alrededor de 2.5 km de longitud. Aunque el modelo costó 250.000 dólares, su nivel de detalle fue tan extremo que nunca se pudo ejecutar debido a los enormes recursos de memoria que requería.

La práctica recomendada es desarrollar el modelo de manera iterativa, comenzando con una versión simple que puede ampliarse gradualmente. Dividir el sistema en submodelos y mantener un nivel uniforme de complejidad facilita este proceso. Se puede iniciar con un modelo básico para discutir y luego ir refinándolo, aumentando su nivel de detalle. Al comparar los resultados con la versión más simple, se puede evaluar el impacto de cada refinamiento. Eventualmente, cuando los refinamientos adicionales tengan un efecto insignificante en las conclusiones, se puede concluir que no es necesario incorporarlos.

Se pueden aplicar múltiples técnicas para obtener un modelo simplificado de un sistema, entre las cuales se encuentran:

- La omisión de ciertos detalles del modelo implica identificar los factores críticos que determinan el comportamiento del sistema y omitir los menos relevantes. Esta estrategia se justifica porque solo al omitir los factores menos importantes se puede construir un modelo útil. El aumento en el nivel de detalle debe hacerse considerando el impacto que tiene sobre los resultados del estudio, centrándose en representar los aspectos que más influencia tienen.
- La agregación de procesos implica combinar varios procesos del sistema en uno equivalente. Ejemplos comunes incluyen:
  - En la agregación temporal, se agrupa un intervalo de tiempo como una unidad. Por ejemplo, todos los eventos de un día se consideran simultáneos en un instante específico, como las 12 del mediodía.
  - En la agregación de recursos, varios recursos se consideran como una unidad. Por ejemplo, si un operario realiza una serie de tareas consecutivas en una cadena de montaje, se puede modelar la operación completa como un proceso único en lugar de modelar cada tarea individualmente. El Teorema del Límite Central ayuda a determinar la distribución de probabilidad resultante de un proceso agrupado. Si la variable aleatoria del proceso agrupado es la suma de variables aleatorias de subprocesos independientes, entonces la variable del proceso agrupado se aproxima a una distribución normal cuando el número de subprocesos es grande, independientemente de las distribuciones de probabilidad de las variables de los subprocesos.
- El tercer tipo de simplificación implica reemplazar un proceso complejo por otro más simple que se comporte de manera equivalente. Esto puede implicar representar un proceso complejo con una distribución de probabilidad o modelar las demandas de un cliente como eventos distribuidos aleatoriamente, en lugar de modelar toda la estructura de negocio del cliente.

Además de las técnicas mencionadas, se suelen aplicar las siguientes aproximaciones:

- Establecer los límites del modelo y su relación con el entorno requiere realizar aproximaciones, como decidir qué factores externos afectan el funcionamiento del modelo (entradas al modelo) y cuáles se ignoran.
- Los fenómenos aleatorios se modelan usando funciones de probabilidad, generalmente distribuciones teóricas, que los representan de forma aproximada.
- En algunos casos, al modelar la interacción entre fenómenos aleatorios, se asume de manera aproximada que dos variables aleatorias son estadísticamente independientes para simplificar el modelo.
- A veces se asume que la distribución de probabilidad de ciertos parámetros del modelo no cambia con el tiempo, lo cual es razonable si la tasa de cambio es insignificante en comparación con el periodo de interés.

Es necesario evaluar el impacto de las aproximaciones de modelado teniendo en cuenta el propósito específico del estudio de simulación:

- Las simplificaciones deben ser evaluadas según su impacto en la precisión de la estimación del comportamiento del sistema.
- En estudios comparativos, es crucial evaluar si las simplificaciones afectan de manera similar a todos los modelos.

Estimar el comportamiento absoluto de un sistema suele requerir modelos más detallados que comparar varios sistemas. Los modelos matemáticos tienden a ser más precisos al comparar sistemas alternativos que al obtener respuestas absolutas.

Diseño de los experimentos

El diseño de los experimentos ocurre en dos etapas: primero, antes de finalizar el diseño del modelo, se eligen las medidas de comportamiento, los factores a variar y sus niveles. Luego, una vez desarrollado, verificado y validado el modelo, se revisa el diseño experimental con base en el conocimiento adquirido en todo el proceso.

Se debe tener en cuenta las siguientes consideraciones:

- El diseño del experimento determina el tipo de análisis estadístico que debe aplicarse a los resultados.
- Los experimentos de simulación deben ser diseñados para obtener la mayor cantidad de información sobre el comportamiento del sistema con el menor costo posible, tanto en términos de tiempo y esfuerzo del experimentador como en tiempo de computación.

Datos de entrada

La selección de los datos de entrada para la simulación es crucial y a menudo consume más tiempo que el diseño y la programación del modelo en sí, ya que depende en gran medida del éxito del estudio.

Cuando se diseña un modelo estocástico de simulación, es crucial decidir si ciertos aspectos del sistema se representan de manera estocástica o determinista, lo que puede llevar a:

- Cuando se dispone de datos experimentales del sistema, es necesario decidir si durante la simulación se extraen directamente de estos datos o se muestrean a partir de una distribución de probabilidad ajustada.
- Cuando no es posible obtener datos experimentales, ya sea porque el sistema aún no existe o no se pueden recolectar, se puede recurrir a consideraciones teóricas y estimaciones proporcionadas por expertos familiarizados con el sistema.

Traducción del modelo

Se han introducido dos metodologías para describir formalmente modelos: el modelado orientado a la planificación de eventos y el modelado orientado a los procesos. Cada una cuenta con herramientas específicas de software para su aplicación.

Por ejemplo, algunas de las herramientas software que facilitan la aplicación del formalismo DEVSson ADEVS, CD++, DEVS/ C++, DEVSJAVA, DEVSim++, JDEVS, PowerDEV8, Python DEVS, SmallDEV8, etc.

Verificación y validación del modelo

La verificación asegura la precisión de la codificación del modelo, mientras que la validación evalúa si el modelo representa adecuadamente la realidad para los objetivos del estudio. Aunque conceptualmente distintas, su análisis a menudo se entrelaza. Si los resultados de la simulación parecen incorrectos, es importante investigar si se deben a errores de codificación o a hipótesis de modelado inadecuadas.

Algunos de los procedimientos que se emplean son:

- Veriñcación manual de la lógica: se ejecuta la simulación durante un periodo de tiempo corto y se comprueba manualmente los resultados obtenidos.

- Comprobación submodelo a submodelo: se trata de verificar individualmente que cada submodelo produce los resultados esperados para todos los posibles tipos de entradas.
- Comprobación con soluciones conocidas: se ajusta el modelo de modo que represente un sistema de solución conocida y comparar ésta con los resultados de la simulación.
- Test de sensibilidad: se puede modificar el valor de un parámetro, dejando los demás fijos, con el fin de medir la sensibilidad del modelo respecto a ese parámetro. La comparación de la sensibilidad observada en las simulaciones, con la que sería de esperar en el sistema real, puede proporcionar pistas útiles.

Respecto a la validación de los modelos, puede considerarse que ésta tiene tres vertientes diferentes:

1. Si la estructura del modelo representa adecuadamente al sistema real.
2. Si los datos calculados al simular el modelo (el comportamiento del modelo) reproducen de forma adecuada el comportamiento del sistema real.
3. La validación del modelo se ve facilitada cuando el usuario final participa en todas las etapas del diseño. Un ejemplo ilustrativo proviene de un caso donde un analista trabajó en un modelo durante seis meses sin interactuar con el solicitante. En la presentación de resultados, el solicitante rápidamente señaló que el modelo no abordaba su problema específico, lo que destaca la importancia de la colaboración directa con el usuario final.

La validez del modelo se evalúa en función de su propósito específico, siendo un proceso continuo durante su diseño, desarrollo y uso. La confianza en el modelo aumenta a medida que supera pruebas y coincide más con el comportamiento del sistema real. La verificación y validación del modelo son procesos en curso que nunca concluyen realmente.

En la validación del modelo, el enfoque clave radica en determinar si las diferencias entre el modelo y el sistema real son lo suficientemente relevantes como para influir en las conclusiones derivadas del uso del modelo. El objetivo del ingeniero es desarrollar modelos útiles en un tiempo y coste razonables.

Experimentación y análisis de los resultados

En modelos estocásticos, la planificación del experimento y el análisis de resultados se llevan a cabo mediante técnicas estadísticas. Existen dos tipos de simulación: "con terminación" y "sin terminación". La elección entre estos tipos depende del objetivo del estudio, no de la naturaleza del sistema. Las características son:

- Simulación con terminación: La simulación tiene terminación cuando un evento marca su fin de manera natural. Su objetivo es estudiar cómo evoluciona el sistema desde ciertas condiciones iniciales hasta que se cumple una condición de finalización.
- Simulación sin terminación: La simulación es sin terminación cuando no hay un evento natural que la concluya. Su propósito es examinar el comportamiento del sistema una vez que alcanza un régimen de funcionamiento independiente de las condiciones iniciales, es decir, el estado estacionario.

Los resultados de un modelo de simulación se tratan como una muestra de datos y pueden analizarse utilizando inferencia estadística. Sin embargo, muchos tests estadísticos requieren que los datos sean independientes, lo que a menudo no ocurre en los datos simulados. Por lo tanto, antes de sacar conclusiones válidas, se deben aplicar procedimientos para abordar la correlación entre los datos. Además, el tamaño de la muestra (duración de la simulación o número de réplicas) debe ser lo suficientemente grande para proporcionar un nivel de confianza adecuado en la estimación del comportamiento.

Documentación y aplicación de los resultados

Una vez finalizados los pasos de diseño, desarrollo, ejecución de la simulación y análisis de resultados, los últimos elementos del estudio de simulación son la aplicación de los resultados y la documentación. Un proyecto de simulación no se considera completado con éxito hasta que los resultados sean entendidos, aceptados y utilizados.

Es fundamental documentar las conclusiones del estudio de manera clara, concisa y convincente, abordando todos los aspectos importantes para el cliente. Una presentación de resultados descuidada puede hacer que las conclusiones no sean aplicadas, lo que resultaría en un fracaso del proyecto. La presentación de los resultados es tan crítica como las etapas previas del estudio y requiere cuidadosa planificación y diseño.

Además de los resultados del estudio, es crucial documentar el desarrollo y la operación del modelo. Esto puede prolongar su vida útil y aumentar la probabilidad de que las recomendaciones basadas en él sean aceptadas. Una documentación adecuada simplifica las modificaciones y garantiza la utilidad del modelo incluso en ausencia de sus creadores.

Análisis de datos: Introducción a R

El análisis de datos es crucial en un proyecto de simulación. Se requiere analizar los datos del sistema para modelar su comportamiento, así como los datos generados por la simulación del modelo para extraer conclusiones.

Algunos entornos de simulación incluyen herramientas para analizar datos, como representaciones gráficas y ajustes de distribuciones de probabilidad. Sin embargo, muchos entornos gratuitos no ofrecen estas herramientas, por lo que es necesario utilizar herramientas externas para el análisis y modelado de los datos.

El lenguaje R y su plataforma son herramientas gratuitas y de código abierto disponibles para Windows, Mac OS X y Linux. Se pueden descargar desde el Comprehensive R Archive Network (CRAN) en http://cran.r-project.org/.

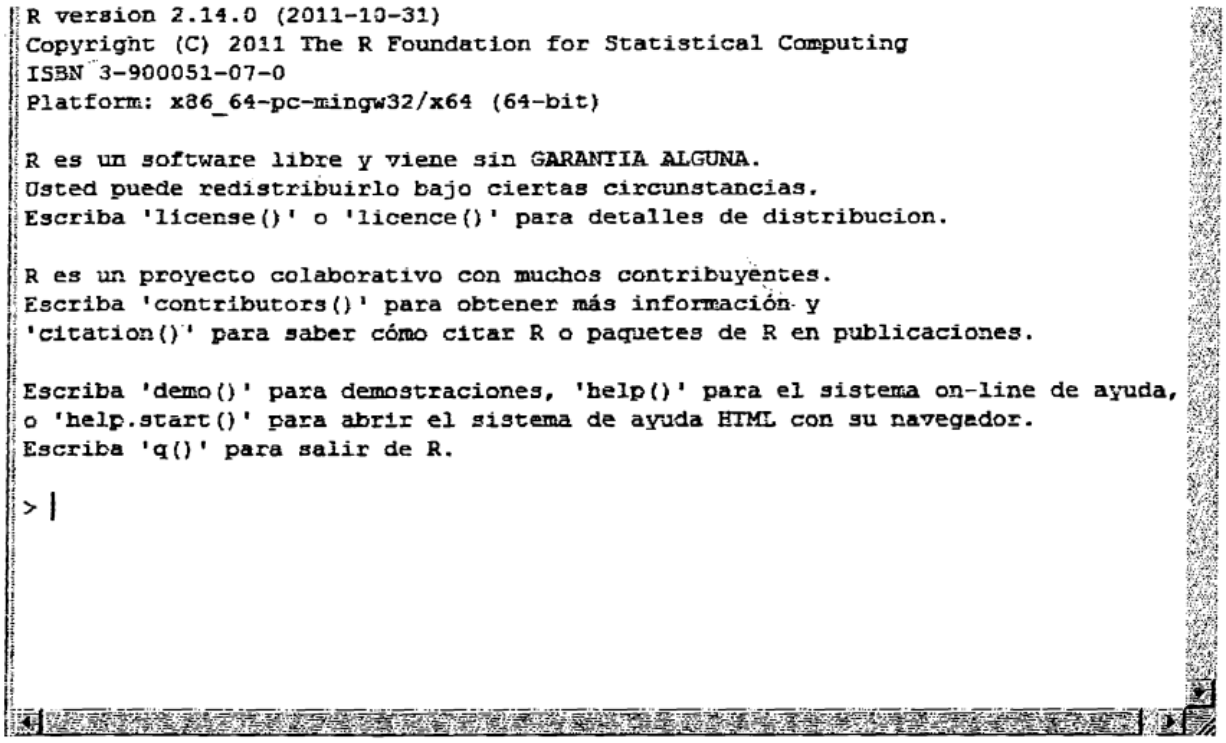

Figura 1.13: Ejemplo de la consola de R en Windows.

Figure 24: Im15

R es un lenguaje interpretado que distingue entre mayúsculas y minúsculas. Permite ingresar comandos uno

por uno en la línea de comandos de la consola, precedidos por el símbolo de sistema (>), o ejecutar conjuntos de comandos desde un archivo. La Figura 1.13 muestra un ejemplo de la consola de R en Windows.

R ofrece una variedad de tipos de datos como vectores, matrices, data frames y listas. Las funciones, tanto predefinidas como definidas por el usuario, son fundamentales en R. Algunas funciones básicas están disponibles automáticamente, mientras que otras se encuentran en paquetes que deben abrirse (conocido como "attach") para acceder a sus funciones.

Las sentencias consisten en funciones y asignaciones. R usa el símbolo <- para las asignaciones, en lugar del típico símbolo =. Por ejemplo,

x < rnorm (10)

crea un objeto de tipo vector llamado x, que contiene 10 observaciones de la distribución normal estándar.

Los comentarios son precedidos por el símbolo #, se ignora todo el texto que aparezca tras el símbolo #. Para finalizar una sesión debe ejecutarse la función q().

La función c() convierte sus argumentos en un vector o una lista. Por ejemplo, supongamos que se realizan 5 réplicas independientes de la simulación del modelo de la oficina, obteniéndose los siguientes 5 valores del tiempo medio de espera en cola: 1.2, 2.4, 1.4, 2.2 y 3.2 minutos. La primera de las siguientes dos sentencias crea un objeto del tipo vector llamado tCola, cuyos componentes son esos cinco valores. La segunda sentencia dibuja el valor del componente del vector frente al índice. En la Figura 1.14 se muestra el gráfico.

tCola <— c(1.2, 2.4, 1.4, 2.2, 3.2) plot(tCola)