



UNIVERSIDAD DE BELGRANO

Las tesinas de Belgrano

**Facultad de Ingeniería y Tecnología Informática
Carrera de Ingeniería Informática**

**Simulación: Aplicación de técnicas de
simulación utilizando un lenguaje específico
(Extend)**

Nº 134

Martín Barbis

Tutor: Darío Piccirilli

Departamento de Investigación
Febrero 2005

Indice

1. Introducción	5
1.1 Objetivo	5
1.2 Alcance	5
1.3 Organización del Trabajo	5
2. Simulación	5
2.1 Conceptos Introdutorios	5
2.2 El Proceso de Simulación	7
2.3 Simulación de Monte Carlo	10
2.3.1 Conceptos Generales	10
2.3.2 Ejemplo	10
2.4 Validación de Modelos de Simulación	16
2.4.1 Introducción	16
2.4.2 Validación de modelos de simulación	16
2.4.3 Validación de los resultados de un modelo	16
2.4.4 Conclusiones	17
3. Lenguajes de Simulación	17
3.1 Introducción	17
3.2 Ventajas y desventajas de los lenguajes de simulación	18
3.3 Clasificación del software de simulación	19
3.4 Características deseables de un lenguaje de simulación	19
3.4.1 Características generales	19
3.4.2 Animación	19
3.4.3 Capacidades estadísticas	20
3.4.4 Soporte	20
3.4.5 Reportes	20
4. Extend	20
4.1 Introducción	20
4.2 Versiones	21
4.2.1 Extend CP (Continuous Process)	22
4.2.2 Extend OR (Operations Research)	22
4.2.3 Extend Industry (Industrial Strength)	22
4.2.4 Extend Suite (Total Simulation Project Support)	22
4.2.5 Tipos de Uso y Costos del Software	23
4.3 Ambiente de Trabajo	23
4.3.1 Bloques	25
4.3.2 Librerías	25
4.3.3 Cuadros de diálogo	26
4.3.4 Conectores y conexiones	26
4.3.5 Menús y barras de herramientas	26
4.4 Librerías	27
4.4.1 Librerías Internas	27
4.4.2 Librerías Externas	27
4.4.3 Uso de Librerías	28
4.5 Entrada y Salida	28
4.5.1 Plotters	28
4.5.2 Notebook del modelo	29
4.5.3 Imprimir	30
4.5.4 Copiar y Pegar, Drag & Drop	31
4.5.5 Datos compartidos dentro del modelo usando Global Array Manager	31
4.5.6 Acceso a datos mediante archivos locales o remotos	32
4.5.7 Acceso a datos mediante bloques de comunicación a bases de datos ODBC/SQL	32
4.5.8 Comunicación entre procesos	33
4.5.9 Objetos OLE/COM y controles ActiveX embebidos	33
4.5.10 Reportes	33
4.6 Lenguaje de Programación	34

4.6.1 Partes de un bloque	35
4.6.2 Introducción al código ModL	38
4.7 Soporte	40
5. Caso de Estudio	41
5.1 Solución en Forma Manual	42
5.2 Solución Utilizando Excel	44
5.4 Comparación de Resultados	56
6. Conclusiones	56
7. Anexo A: Tabla de Números Pseudo-Aleatorios	57
8. Anexo B: Tabla de Lenguajes de Simulación	58
9. Anexo C: Extend: Características Avanzadas	71
9.1 Extend CP	71
9.1.1 Clonación de ítems	72
9.1.2 Animación	72
9.1.3 Análisis de sensibilidad	73
9.1.4 Optimización	74
9.1.5 Edición de ecuaciones	75
9.1.6 Bloques de control	76
9.2 Extend OR	77
9.2.1 Discrete Event	77
9.2.2 Manufacturing	77
9.2.3 QuickBlocks	78
9.3 Extend Industry	78
9.3.1 Base de datos integrada	79
9.3.2 Arquitectura de flujo	79
9.4 Extend Suite	79
9.4.1 Stat::Fit	79
9.4.2 Proof Animation	80
10. Anexo D: El Lenguaje ModL	81
10.1 Estructura	81
10.1.1 Nombres	81
10.1.2 Tipos de datos	81
10.1.3 Conversiones de tipos	82
10.1.4 Vectores	82
10.1.5 Operadores	82
10.1.6 Expresiones de control	83
10.1.7 Procedimientos y funciones definidas por el usuario	83
10.2 Manejadores de mensajes	83
10.3 Variables de sistema	86
10.4 Variables globales	86
10.5 Vectores globales	87
10.6 Listas encadenadas	87
10.7 Constantes	87
10.8 Funciones	87
11. Anexo E: Bibliografía	88

Introducción

Objetivo

El objetivo de esta tesina es realizar una investigación acerca del lenguaje de simulación Extend, que está orientado a la creación de modelos de procesos industriales o de servicios. El trabajo también comprenderá el desarrollo de un modelo para simular un sistema en particular en el lenguaje seleccionado.

Alcance

El trabajo comprende:

- Una introducción a la simulación.
- Una descripción de las características de los lenguajes de simulación.
- Un análisis del lenguaje de simulación Extend.
- La implementación de un modelo en el lenguaje elegido.

Organización del Trabajo

El presente trabajo está organizado de la siguiente manera:

- Capítulo 1: Consta de la introducción, donde se describe el objetivo y la organización del trabajo.
- Capítulo 2: Se analiza el concepto de Simulación, lo cual sirve como base para los capítulos siguientes.
- Capítulo 3: Reúne información acerca de las características de los lenguajes de simulación.
- Capítulo 4: Describe detalladamente el lenguaje de simulación Extend.
- Capítulo 5: Se presenta un problema y se lo resuelve mediante la implementación de un modelo. Se muestra la solución en el lenguaje seleccionado.
- Capítulo 6: Se desarrollan las conclusiones y se analizan posibles extensiones a esta tesina.

El Anexo A consta de una tabla de números pseudo-aleatorios. En el Anexo B se encuentra una tabla con los principales lenguajes de simulación y algunas de sus características. El Anexo C describe las características avanzadas de Extend. El Anexo D describe más detalladamente el lenguaje ModL, incluido en Extend. El Anexo E detalla la bibliografía utilizada.

Simulación

2.1. Conceptos Introductorios

La palabra Simulación, en su concepto actual, se remonta hacia fines de 1940 cuando Von Neumann y Ulam acuñaron el término «análisis de Monte Carlo» para aplicarlo a una técnica matemática que usaban entonces para resolver ciertos problemas de protección nuclear que eran, o demasiado costosos para resolverse experimentalmente o demasiado complicados para ser tratados analíticamente.

Con el advenimiento de la computadora de gran velocidad, a principios de 1950, la simulación tomó otro significado, ya que surgió la posibilidad de experimentar con modelos matemáticos (que describen sistemas de interés) en una computadora.

Una definición de simulación, estrictamente formal, es la propuesta por C. West Churchman, la cual admite las ambigüedades e inconsistencias inherentes al uso actual de la palabra:

«**x** simula a **y**» si y sólo si: (a), **x** e **y** son sistemas formales; (b), **y** se considera como el sistema real; (c), **x** se toma como una aproximación del sistema real; (d), las reglas de validez en **x** no están exentas de error.¹

Si bien la definición de Shubik no es tan precisa, parece más apropiada:

«Simulación de un sistema (o de un organismo) es la operación de un modelo (simulador), el cual es una representación del sistema. Este modelo puede sujetarse a manipulaciones que serían imposibles de realizar, demasiado costosas o impracticables. La operación de un modelo puede estudiarse y con ello, inferirse las propiedades concernientes al comportamiento del sistema o subsistema real»²

Tomando como principal interés los experimentos de simulación que pueden realizarse en una computadora digital, la definición puede reformularse de la siguiente manera:

Simulación es una técnica numérica para conducir experimentos en una computadora digital, los cuales requieren de ciertos tipos de modelos lógicos y matemáticos, que describen el comportamiento de un sistema o de algún componente de él, en períodos de tiempo real.

Un modelo de simulación busca imitar el comportamiento del sistema que investiga estudiando las interacciones entre sus componentes. Es el seguimiento a lo largo del tiempo de los cambios que tienen lugar en el modelo dinámico del sistema.

La simulación de un sistema permite reunir información acerca del comportamiento del mismo, mediante la ejecución de un modelo computarizado.

Un modelo es una descripción lógica de cómo un sistema, proceso o componente se comporta. Mediante un modelo se puede experimentar y probar alternativas sin incurrir en los costos que implica utilizar el sistema real, que inclusive podría no existir aún.

La simulación se enmarca dentro de la Investigación de Operaciones, definiendo la misma como: *El uso del método científico para proveer criterios para la toma de decisiones en los sistemas hombre-máquina.*

Es decir que la Investigación Operativa le permite a quien toma las decisiones seleccionar la mejor alternativa dentro del conjunto que ha sido estudiado.

El fundamento racional para usar la simulación en cualquier disciplina es la búsqueda constante del hombre por adquirir conocimientos relativos a la predicción del futuro.

La simulación ha demostrado ser útil en muchas áreas de la ciencia y la tecnología, incluyendo tanto aplicaciones científicas como prácticas, por ejemplo:

Problemas teóricos de las ciencias básicas: matemáticas, física, y química:

- o Estimación del área encerrada por una curva.
- o Inversión de matrices.
- o Estimar la constante p .
- o Solución de ecuaciones.

Problemas prácticos en todos los aspectos del mundo real:

- o Simulación industrial: Diseño de procesos químicos, control de inventarios, diseño de sistemas de distribución, programación de mantenimiento, diseño de sistemas de comunicaciones, etc.
- o Simulación de problemas comerciales y económicos: Operación de toda la compañía, conducta del cliente, evaluación de gastos de capital propuestos, determinación de precios, propuestas de mercado, estudio de economías nacionales con problemas de recesión e inflación, planes de desarrollo, políticas de balance de pagos de economías subdesarrolladas, predicción económica, etc.
- o Problemas conductuales y sociales: Dinámica de población, conducta individual y de grupo.
- o Simulación de problemas biomédicos: Equilibrio de líquidos, distribución de electrolitos en el cuerpo humano, representación del cerebro a través de modelos, proliferación de células sanguíneas, difusión de epidemias, etc.
- o Estrategias y tácticas de guerra.

A pesar de que los modelos de simulación simples pueden ser resueltos manualmente, la simulación de problemas prácticos requiere del uso de programas de computadoras.

Si bien en un principio los modelos se desarrollaban en lenguajes de programación generales, pronto surgieron herramientas que facilitaban su construcción. Estas herramientas son específicas para simulación. Entre ellas existen aplicaciones de propósito general, aplicaciones para áreas específicas, lenguajes de simulación y simuladores gráficos.

La tendencia actual en cuanto a herramientas de simulación es la de combinar la facilidad de uso de un simulador gráfico con la flexibilidad de un lenguaje. A estas herramientas se las llama híbridos.

Independientemente del tipo de herramienta que se utilice, existen dos tipos de modelos de simulación, los modelos continuos y los discretos.

Los modelos continuos tratan con sistemas cuyo comportamiento cambia continuamente de forma con el tiempo. Ejemplos de sistemas continuos son los simuladores de vuelo con los que las aerolíneas entrenan pilotos, los sistemas de simulación que el servicio meteorológico utiliza para predecir futuros patrones de clima, y las simulaciones que permiten a ingenieros de proceso predecir las características operativas de proyectos.

Los modelos discretos tratan con sistemas cuyo comportamiento sólo cambia en instantes dados. Un ejemplo típico ocurre en las líneas de espera donde se desea estimar medidas como el tiempo de espera promedio o la longitud de la línea de espera. Los momentos en los que ocurren los cambios en el sistema identifican los eventos del modelo. El hecho de que los eventos ocurren en puntos discretos da lugar al nombre de simulación de eventos discretos.

También existen los modelos híbridos de sistemas continuos y discretos. Por ejemplo, al simular el inventario de combustible en un aeropuerto, los clientes (aviones) llegan como eventos discretos, mientras que el combustible que ingresa en cada avión es un flujo continuo.

Factor	Modelos de sistemas continuos	Modelado de sistemas discretos
¿Qué se está modelando?	Flujos.	Ítems.
Características	Números aleatorios simulan características de flujos y deben ser repetidos para cada consulta.	Las características son asignadas a ítems mediante atributos y prioridades que pueden ser monitoreadas a través del modelo
Intervalo de tiempo	Los intervalos de tiempo son generalmente constantes. Los recálculos del modelo son secuenciales y dependientes del tiempo.	Los intervalos de tiempo depende de cuando ocurren los eventos. El modelo sólo recalcula cuando ocurren eventos.
Orden	Los flujos siguen un orden FIFO ³ .	Los ítems pueden tener flujos FIFO, LIFO ⁴ , por prioridad, o personalizado.
Encaminamiento	Los flujos deben ser encaminados siendo apagados en un punto y encendidos en otro. Los flujos pueden ir a varios lugares al mismo tiempo.	Los ítems son encaminados automáticamente al primer punto disponible. Los ítems solo pueden estar en un lugar en un momento dado.
Detalle estadístico	Sólo estadísticas generales acerca del sistema: cantidades, eficiencia, tiempo de tránsito.	Además de las estadísticas generales, cada ítem puede ser individualmente monitoreado: conteo, utilización, tiempo del ciclo.
Usos comunes	Ciencia (biología, química, física), Ingeniería (electrónica, sistemas de control), Economía, Dinámica de sistemas.	Manufactura, Industrias de servicios, Reingeniería de procesos de negocios, Planeamiento estratégico, Redes, Ingeniería de sistemas.

Tabla 2.1: Comparación de modelos de sistemas continuos y discretos

2.2. El Proceso de Simulación

La experiencia sugiere que la planificación de experimentos de simulación requiere de un procedimiento que conste de las etapas siguientes:

1. Formulación del problema
2. Recolección y procesamiento de datos tomados de la realidad.
3. Formulación de un modelo matemático.
4. Estimación de los parámetros de las características operacionales a partir de los datos reales.
5. Evaluación del modelo y de los parámetros estimados.
6. Formulación del programa para la computadora.
7. Validación
8. Diseño de experimentos de simulación
9. Análisis de datos simulados.

A continuación se describen los pasos antes mencionados.

1. Formulación del problema

El estudio de la simulación en computadoras tiene que comenzar con la formulación de un problema o con una declaración explícita de los objetivos del experimento. Es decir, definir claramente los objetivos de la investigación, antes de planificar la realización de cualquier experimento de simulación.

La exposición original del problema varía considerablemente de su versión final, ya que la formulación del problema es un proceso secuencial que generalmente requiere una reformulación continua y progresiva y un refinamiento de los objetivos del experimento durante su realización.

Si el objetivo del estudio de simulación es obtener respuestas a una o más preguntas, es necesario plantear éstas detalladamente desde el comienzo del experimento, aún cuando sea posible redefinir las preguntas durante el curso del mismo. Especificar sólo las preguntas a contestarse no es suficiente para realizar un experimento de simulación, se requiere también establecer los criterios para evaluar las posibles respuestas a estas preguntas.

Por otro lado, el objetivo de los esfuerzos de investigación podría consistir en probar una o más hipótesis relativas al comportamiento del sistema bajo estudio. En cada caso es necesario que las hipótesis que deban probarse se planteen explícitamente, así como los criterios para su aceptación o rechazo.

Finalmente, el objetivo podría consistir en estimar los efectos que ciertos cambios en los parámetros, las características operacionales o las variables exógenas, tendrán sobre las variables endógenas del sistema. Sin embargo, antes de estimar los efectos sobre cualquier variable endógena en un sistema dado, es necesario especificar los requerimientos en términos de precisión estadística.

Por consiguiente, deben tomarse dos decisiones importantes antes de comenzar a trabajar con cualquier experimento de simulación. En primer término, hay que decidir los objetivos de la investigación y, en

segundo lugar, es necesario decidir el conjunto de criterios para evaluar el grado de satisfacción al que deba sujetarse el experimento a fin de cumplir los objetivos.

2. Recolección y procesamiento de datos tomados de la realidad

Es posible identificar seis funciones importantes del procesamiento de datos que forman una parte integral del procedimiento para implantar los experimentos de simulación en computadoras: recolección, almacenamiento, conversión, transmisión, manipulación y salida.

La *recolección* de datos es el proceso de captación de los hechos disponibles, para su procesamiento posterior, cuando sea necesario. En realidad, el proceso de recolección y el de *almacenamiento* de datos ocurren simultáneamente, pues el primero implica que los datos hayan sido almacenados.

La *conversión* de los datos de una forma a otra tiene una función crucial en la determinación de la eficacia del procesamiento.

Bajo ciertas circunstancias, existen problemas adicionales a la conversión de los datos de una forma a la otra, que implican una *transmisión* de ellos, esto es, el transporte de la información desde determinado lugar hasta donde será procesada.

Una vez que los datos han sido recolectados, almacenados, convertidos a una forma eficaz y transmitidos al lugar del procesamiento final, resulta posible entonces, comenzar con las operaciones de manipulación de datos y la preparación de éstos para su salida final.

3. Formulación de los modelos matemáticos

La formulación de modelos matemáticos consiste en tres pasos:

- a. Especificación de los componentes.
- b. Especificación de las variables y los parámetros.
- c. Especificación de las relaciones funcionales.

Consideraciones para la formulación de un modelo matemático:

- Cantidad de variables que se deben incluir en el modelo: Se encuentra poca dificultad en lo referente a variables endógenas o de salida del modelo, debido, por lo general, a que estas variables se determinan al comenzar el experimento, cuando se formulan los objetivos del estudio. La dificultad real surge en la elección de variables exógenas (algunas de las cuales pudieran ser estocásticas) que afectan a las variables endógenas. La existencia de muy pocas variables exógenas puede llevar a modelos inválidos, en tanto que una abundancia de ellas hace, a veces, imposible la simulación en la computadora debido a la complicación que genera en los programas.
- Complejidad de los modelos: Se deben diseñar modelos matemáticos que produzcan descripciones o predicciones, razonablemente exactas, referentes al comportamiento de un sistema dado y reduzcan a la vez, el tiempo de computación y programación.
- Eficiencia de computación: Entendiendo por ello la cantidad de tiempo de cómputo requerida para lograr algún objetivo experimental específico.
- El tiempo requerido para escribir el programa.
- Validación: La cantidad de realismo incorporado en los modelos. Es decir, si el modelo describe adecuadamente el sistema de interés, si proporciona predicciones razonablemente buenas acerca del comportamiento del sistema, etc.

4. Estimación de los parámetros de las características operacionales a partir de los datos reales

Una vez que se han recolectado los datos apropiados del sistema y formulado varios modelos matemáticos que describen su comportamiento, es necesario estimar los valores de los parámetros de dichos modelos y probar su significación estadística.

5. Evaluación del modelo y de los parámetros estimados

Es necesario hacer un juicio del valor inicial de la suficiencia del modelo una vez que se formulan el conjunto de modelos matemáticos que describen el comportamiento del sistema y que se estimaron los parámetros de sus características operacionales sobre la base de las observaciones tomadas del mundo real; es decir, se debe probar el modelo.

Más adelante se considerará la validez de los datos de salida generados por el modelo de simulación. En tal caso, interesará comparar los datos simulados con datos históricos, para así determinar la capacidad predictiva del modelo.

6. Formulación de un programa para la computadora

La formulación de un programa para computadora, cuyo propósito sea dirigir los experimentos de simulación, requiere que se consideren especialmente las siguientes actividades:

- a. Diagrama de flujo
- b. Lenguaje de desarrollo
 - Compiladores de propósitos generales
 - Lenguajes de simulación de propósitos especiales
- c. Búsqueda de errores
- d. Datos de entrada y condiciones iniciales
- e. Generación de datos
- f. Reportes de salidas

Al escribir un programa de simulación para computadora, la primera etapa requiere la formulación de un diagrama de flujo que bosqueje la secuencia lógica de los eventos que realizará la computadora, al generar los tiempos planificados para las variables endógenas del modelo.

Una vez finalizado el diagrama de flujo, se deberá escribir el código para la computadora, que se utilizará en las corridas de los experimentos, para lo cual se dispone de dos alternativas:

- Escribir el programa en un lenguaje de propósitos generales como el FORTRAN, PL/I, Pascal, C, o C++.
- Emplear un lenguaje de simulación de propósitos especiales como el GPSS, SIMSCRIPT, SIMAN, SLAM, Alpha/Sim, SIMPROCESS, ProModel o Extend.

El ahorro en tiempo de programación constituye la principal ventaja al utilizar un lenguaje de simulación de propósitos especiales, en lugar de un compilador de propósitos generales, ya que dichos lenguajes fueron escritos para facilitar la programación de ciertos tipos de sistemas. Pero aunque existe la posibilidad de reducir el tiempo de programación mediante el empleo de un lenguaje de simulación, se reduce de esta manera la flexibilidad de los modelos y se incrementa el tiempo de cómputo.

Otra ventaja importante de los lenguajes de simulación de propósitos especiales, consiste en que usualmente proporcionan técnicas para la búsqueda de errores.

Un aspecto más en la fase de programación del desarrollo de un experimento de simulación en computadora, se refiere a los datos de entrada y las condiciones iniciales para el experimento. Como los experimentos de simulación son dinámicos, surge una pregunta respecto al valor que se les debería asignar a las variables y parámetros del modelo en el momento que se comienza a simular el sistema, es decir, el desarrollo de las técnicas numéricas para la generación de datos.

Presuponiendo que una o más de las variables exógenas, incluidas en los modelos matemáticos, son variables estocásticas que tienen una distribución conocida de probabilidad, se debe diseñar el proceso aleatorio de selección a partir de una distribución dada de probabilidad, tal que, los resultados de la repetición de este proceso en una computadora digital, originen una distribución de probabilidad de los valores muestreados correspondiente a la distribución de la variable de interés.

Los reportes de salida, necesarios para obtener información relativa al comportamiento del sistema bajo simulación, constituyen una consideración final en el desarrollo del proyecto. Si se usa un lenguaje de propósitos generales, existirá un mínimo de restricciones impuestas sobre el formato de los reportes de salida. Sin embargo, si se utiliza un lenguaje de simulación de propósitos especiales, es imprescindible ajustarse a los requisitos en el formato de salida, impuestos por este lenguaje.

7. Validación

Ciertamente, el problema de validar modelos de simulación es difícil ya que implica un sinnúmero de complejidades de tipo práctico, teórico, estadístico e inclusive filosófico. La validación de experimentos de simulación forma parte de un problema mucho más general, es decir, el de la validación de cualquier clase de modelo o hipótesis.

Por lo general sólo dos pruebas se consideran apropiadas para validar los modelos de simulación:

- La coincidencia de los valores simulados de las variables endógenas con datos históricos conocidos, si es que estos están disponibles
- La exactitud de las predicciones del comportamiento del sistema real hechas por el modelo de simulación, para períodos futuros. Asociadas con estas pruebas, existe una gran variedad de herramientas estadísticas.

8. Diseño de los experimentos de simulación

Una vez validado el modelo para la computadora, se puede utilizar en un experimento de simulación.

El interés se centra ahora en el diseño experimental.

En esta fase es posible identificar dos metas importantes: en primer lugar, la selección de los valores de los factores (variables exógenas y parámetros) y las combinaciones de valores, así como el orden de los

experimentos; una vez seleccionadas las combinaciones de factores, el esfuerzo se pondrá en asegurar que los resultados queden razonablemente libres de errores fortuitos.

9. Análisis de los datos simulados

La etapa final en el procedimiento requiere un análisis de los datos generados por la computadora, a partir del modelo que se simula. Tal análisis comprende tres pasos:

- a. Recolección y procesamiento de datos simulados
- b. Cálculo de la estadística de las pruebas
- c. Interpretación de los resultados

Aún cuando el análisis de los datos simulados es, de hecho, semejante al análisis de los datos del mundo real, existen algunas diferencias importantes. Teichroew⁶ ha señalado, que al comparar la simulación en computadora con la técnica estadística conocida por el nombre de muestreo de distribuciones, el análisis de los datos de simulación en computadora es, considerablemente, más difícil que el análisis de los datos del mundo real.

2.3. Simulación de Monte Carlo

2.3.1. Conceptos Generales

Si el sistema que se está simulando incluye entradas que son variables aleatorias, el modelo de simulación debería reflejarlas con la mayor precisión posible. Una forma de hacerlo es usando una técnica llamada simulación de Monte Carlo, en la cual el simulador se diseña para que los eventos simulados ocurran aleatoriamente y reflejen las frecuencias teóricas que se están modelando.

El método fue llamado así por el principado de Mónaco por ser la «capital del juego de azar», al tomar una ruleta como un generador simple de números aleatorios. El nombre y el desarrollo sistemático de los métodos de Monte Carlo datan aproximadamente de 1944 con el desarrollo de la computadora electrónica. Sin embargo hay varias instancias (aisladas y no desarrolladas) en muchas ocasiones anteriores a 1944.

La simulación de Monte Carlo fue creada para resolver integrales que no se pueden resolver por métodos analíticos; para resolver estas integrales se usaron números aleatorios. Posteriormente se utilizó para cualquier esquema que emplee números aleatorios, usando variables aleatorias con distribuciones de probabilidad conocidas.

2.3.2. Ejemplo

La simulación de Monte Carlo utiliza números aleatorios, que pueden generarse por programas de computación o tomados de una tabla de números aleatorios, para generar eventos simulados. El proceso de relacionar números aleatorios con eventos simulados se llama mapeo de números aleatorios. Para ilustrar como se desarrollan los mapeos de números aleatorios y cómo se usan los números aleatorios en una simulación de Monte Carlo, consideremos el siguiente ejemplo de la Jewel Vending Company⁶.

1. Problema

Bill Jewel es el dueño de la Jewel Vending Company, que instala y provee máquinas expendedoras de golosinas en supermercados, tiendas y restaurantes. Bill está considerando instalar una máquina expendedora de caramelos en un supermercado de la Avenida Lincoln. La máquina expendedora contiene 80 caramelos. Idealmente, Bill desea llenar la máquina cada vez que su contenido llega a la mitad.

Basado en la performance de lugares similares, Bill estimó la siguiente distribución de demanda diaria:

- $P(\text{demanda diaria} = 0 \text{ caramelos}) = .10$
- $P(\text{demanda diaria} = 1 \text{ caramelo}) = .15$
- $P(\text{demanda diaria} = 2 \text{ caramelos}) = .20$
- $P(\text{demanda diaria} = 3 \text{ caramelos}) = .30$
- $P(\text{demanda diaria} = 4 \text{ caramelos}) = .20$
- $P(\text{demanda diaria} = 5 \text{ caramelos}) = .05$

Bill desea estimar el número de días que le tomará a una máquina llena llegar a 40 unidades. Esta información le permitirá saber la frecuencia para rellenar la máquina.

2. Solución

Bill puede considerar estimar el tiempo esperado entre rellenos calculando el promedio de demanda diaria, basado en la distribución de probabilidad y dividiendo 40 por este valor.

$$\text{Tiempo esperado} = \frac{40}{.1(0) + .15(1) + .2(2) + .3(3) + .2(4) + .05(5)} = \frac{40}{2.5} = 16 \text{ días}$$

Bill no está seguro de que este método le de el verdadero promedio de días requeridos para vender 40 o

más caramelos, pero considera que es una buena aproximación. Para probarla utilizará la simulación.

3. Simulación de demanda diaria

Para simular el sistema de Jewel Vending Company, definimos la variable aleatoria:

X = demanda diaria de caramelos en el supermercado

Con base en las estimaciones de Bill, la función de distribución de probabilidad para X es la de la siguiente tabla:

x	$P(X = x)$
0	.10
1	.15
2	.20
3	.30
4	.20
5	.05

Tabla 2.2: Distribución de probabilidad

La teoría aplicada supone que si se simulan 1000 días del sistema de Jewel Vending Company, los resultados esperados son aproximadamente 100 días de demanda nula, 150 días de demanda 1, etc. Sin embargo, no se debe seguir un patrón, como que la demanda sea cero en los días 1, 11, 21, etc.

4. Mapeo de números aleatorios

Una forma de simular eventos aleatorios que siga la distribución de probabilidad deseada es generar números aleatorios entre 00 y 99 de forma tal que cada número tenga igual probabilidad de ser elegido (distribución uniforme). Luego se asignan 10 de estos números al evento «demanda diaria = 0», 15 al evento «demanda diaria = 1», etc. Dado que cada número entre 00 y 99 tiene la misma probabilidad de ser elegido, la probabilidad de cualquier número es de 1/100. Por lo tanto, si 10 de estos números son asignados a «demanda diaria = 0», este evento ocurrirá con una probabilidad de .10.

El modo más simple de hacer esto es asignar los primeros 10 números a «demanda diaria = 0», los siguientes 15 a «demanda diaria = 1», etc. Este proceso se denomina mapeo de números aleatorios, porque relaciona un número aleatorio a la salida de un evento simulado.

Demanda diaria (X)	Números aleatorios correspondientes
0	00-09
1	10-24
2	25-44
3	45-74
4	75-94
5	95-99

Tabla 2.3: Mapeo de números aleatorios

5. Enfoque de distribución acumulada para el mapeo de números aleatorios

A pesar de que este enfoque funciona bien para variables aleatorias discretas, es necesario un enfoque más general para distribuciones continuas. Una forma que puede usarse tanto con distribuciones discretas como continuas involucra el uso de la función de distribución acumulada para la variable aleatoria, X . Una distribución acumulada, $F(x)$, da la probabilidad de que X sea menor o igual a algún valor x ; o sea, $F(x) = P(X \leq x)$. La función de distribución acumulada para el problema de Jewel Vending Company es la siguiente:

x	$F(x) = P(X \leq x)$
0	.10
1	.25
2	.40
3	.70
4	.90
5	1.00

Tabla 2.4: Distribución acumulada

Si un número aleatorio Y (entre 0 y 1) es elegido, se puede determinar el valor para el evento encontrando el valor mínimo de x para que $F(x) \geq Y$. Por ejemplo, si se elige el número 0.34, el valor mínimo de x para que $F(x) \geq 0.34$ es 2. Este es el mismo valor de demanda que se obtenía mediante la tabla de mapeo de números aleatorios.

6. Generación de números aleatorios

Una forma de generar números aleatorios entre 00 y 99 consiste en tomar 100 bolas y marcar cada una de ellas con un número entre 00 y 99 y ponerlas en una caja. Un número aleatorio puede generarse eligiendo una bola de la caja y mirando su número.

En la práctica, lo que se hace es imitar este proceso mediante el uso de una computadora con un generador de números aleatorios. Un generador de números aleatorios en realidad genera lo que se denominan números pseudo-aleatorios, porque los números no son realmente aleatorios sino que son generados a partir de una fórmula matemática.

Un generador de números aleatorios parte de un valor inicial, llamado semilla, y produce una secuencia de números que tienen las siguientes características:

- Todos los números tienen la misma probabilidad de ocurrir.
- No hay relación aparente entre los números generados por la fórmula.

7. Simulación del problema

Usando un conjunto de números pseudo-aleatorios (ver Anexo A) se ilustrará como realizar una simulación para el problema propuesto. El enfoque es el siguiente:

Comenzando en el día 1, un número aleatorio es elegido para determinar la demanda diaria para ese día. El valor de demanda será usado para actualizar el número total demandado hasta la fecha. La simulación será repetida para el día 2, luego el día 3, etc., hasta que la demanda total acumulada alcance 40 o más. El número de días simulados requeridos para alcanzar la demanda de 40 o más luego será almacenado.

Dado que sólo se necesitan números aleatorios de dos dígitos para generar demandas de caramelos, la simulación comienza usando los primeros dos dígitos de la primera fila de la columna 1 en el Anexo A. Para cada día subsiguiente, una nueva demanda será determinada usando el número de dos dígitos de la siguiente fila de la columna 1.

El primer número de la columna 1 es 6506. Los primeros dos dígitos son 65. Según el mapeo de la Tabla 2.3, el número aleatorio correspondiente al número 65 es de tres caramelos. Para el día 2, se utilizan los primeros dos dígitos del número aleatorio de la segunda fila de la primera columna, 77. Esto corresponde a una demanda de cuatro unidades. Continuando en la columna 1, se llega al resultado de la tabla siguiente:

Día	Número aleatorio	Demanda	Demanda total acumulada
1	65	3	3
2	77	4	7
3	61	3	10
4	88	4	14
5	42	2	16
6	74	3	19
7	11	1	20
8	40	2	22
9	03	0	22
10	62	3	25
11	54	3	28
12	10	1	29
13	16	1	30
14	69	3	33
15	16	1	34
16	02	0	34
17	31	2	36
18	79	4	40

Tabla 2.5: Simulación

Día	Número aleatorio	Demanda	Demanda total acumulada
1	42	2	2
2	74	3	5
3	93	4	9
4	84	4	13
5	89	4	17
6	89	4	21
7	12	1	22
8	64	3	25
9	64	3	28
10	38	2	30
11	61	3	33
12	53	3	36
13	12	1	37
14	76	4	41

Tabla 2.6: Segunda simulación

En la primera simulación, se ve que tomó 18 días vender 40 caramelos, 2 días más de los 16 que había estimado Bill originariamente. De todas formas, es difícil sacar conclusiones a partir de una ejecución de simulación. Para conseguir una mejor estimación, es necesario ejecutar varias veces la simulación.

Al ejecutar la simulación una segunda vez, usando como entrada los números aleatorios de la columna 15 del Anexo A, se obtienen los resultados de la Tabla 2.6. Esta vez tomó 14 días vender 40 caramelos. Además la demanda fue de 41 en vez de 40 caramelos.

Se suelen utilizar diagramas de flujo para representar el programa de computación que se utilizará para resolver el problema. La Figura 2.1 muestra un posible diagrama de flujo para la simulación del problema de Jewel Vending Company.

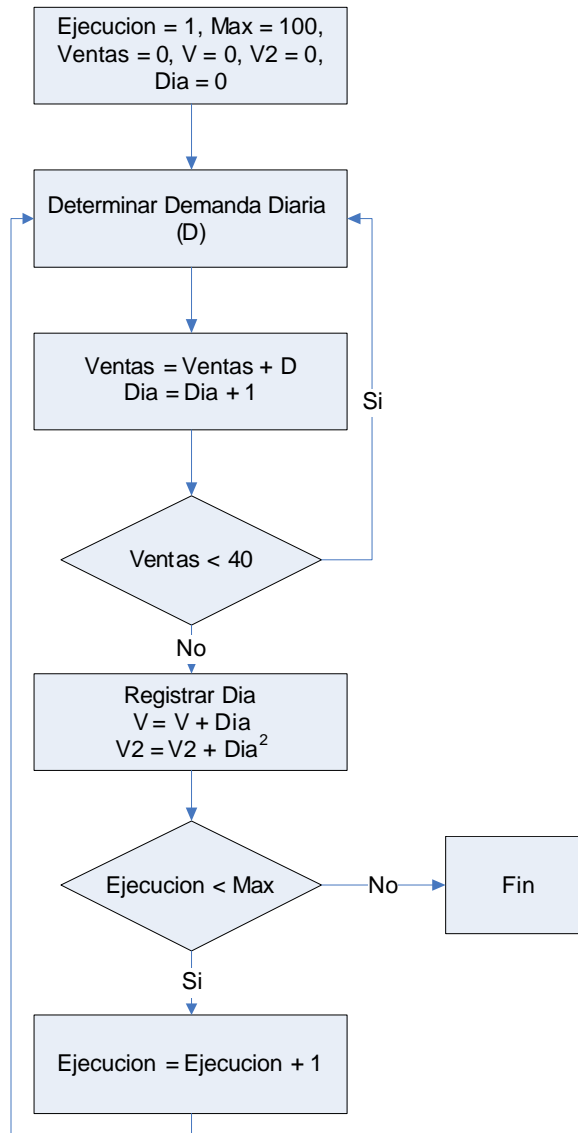


Figura 2.1: Diagrama de Flujo para el problema de Jewel Vending Company

El propósito de ejecutar la simulación, es determinar si el número de días requeridos para vender 40 caramelos es, efectivamente, 16. Ninguna de las dos ejecuciones dio como resultado 16, sin embargo, el promedio del resultado de las dos ejecuciones, 18 y 14, es 16. A medida que se ejecuten más simulaciones, las leyes de probabilidad sugieren que el promedio calculado se acercará más al verdadero promedio.

Se puede utilizar Excel para simular el problema de Jewel Vending Company. Esta simulación se realiza de la manera siguiente:

- Mediante expresiones IF, se imprime el día en la columna A si aún no se vendieron 40 caramelos.
- En la columna B, se genera la demanda diaria usando variables aleatorias y el comando VLOOKUP. La tabla de búsqueda es ingresada en las columnas G y H.
- La demanda acumulada es generada en la columna C, sumando la demanda diaria y la acumulada del día anterior.
- Asumiendo que nunca tomará más de 100 días vender 40 caramelos, las fórmulas de las celdas A6:C6 son arrastradas 100 filas hasta las celdas A105:C105.
- El número máximo de la columna A es el número de días que tomará vender 40 o más caramelos. Este valor se almacena en la celda E1.
- Esta simulación puede ser replicada mediante la tecla F9 en Excel.

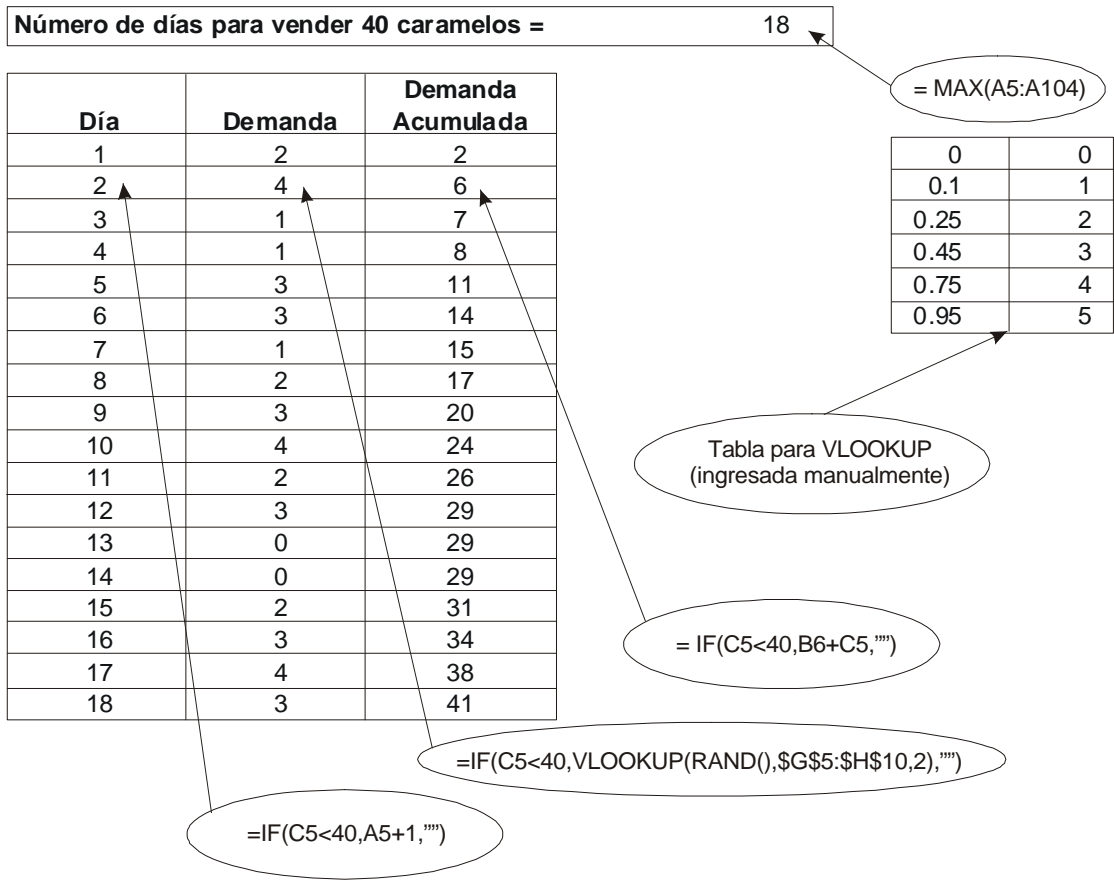


Figura 2.2: Excel. Ejecución de la simulación para el problema de Jewel Vending Company

2.4. Validación de Modelos de Simulación

2.4.1. Introducción

La validación de un modelo de simulación es la demostración de que el modelo representa la realidad con un grado de exactitud adecuado para la aplicación deseada.

Es importante advertir que la validación de modelos no consiste únicamente en procedimientos y construcciones enteramente formales y objetivos. La validación, en cualquier disciplina, debe contener componentes subjetivos y semiformales por varias razones. Una razón importante tiene que ver con la relación entre la validez de un modelo y su propósito. Es imposible desligar el proceso de validación del propósito del modelo. Además, una vez que la validación es vista como «una utilidad con respecto a un propósito», tendría que ser cuestionada la propia «utilidad del propósito». Es decir, juzgar la validez de un modelo fundamentalmente implica juzgar también la validez de su propósito, lo cual es esencialmente un proceso no técnico, informal y cualitativo. Otro aspecto informal y no técnico de la validación es su naturaleza distribuida y prolongada. La validación, para que sea efectiva, debería ser un proceso gradual, repartido a lo largo de toda la metodología de modelado.

2.4.2. Validación de modelos de simulación

La validación de modelos de simulación consta de tres fases, la validación conceptual, la validación lógica y la validación de aptitud.

- Validación conceptual: La validación conceptual se aplica para que en el proceso de conceptualización desde el sistema real a modelar hasta el modelo conceptual, se incluyan y desarrollen las percepciones esenciales de los actores relevantes del proceso. Cabe aclarar que el modelo conceptual es una imagen mental coherente de la situación a resolver, indica la forma en que se encarará el problema de modelado y los elementos que a él pertenecen. Por lo tanto la validación conceptual incluye temas como el proceso de percepción de la situación a resolver, la identificación de los factores a tener en cuenta, la formulación de objetivos y necesidades en los que enfocar el proceso de modelado y validación, la elección de un ángulo de ataque del problema, la obtención de un marco adecuado de desarrollo, y la identificación de elementos clave para la formulación de un modelo formal capaz de obtener soluciones válidas.
- Validación lógica: Los factores que rigen el proceso de transformar el modelo conceptual en un modelo formal constituyen los elementos a examinar por este tipo de validación. Esta transformación, que puede llamarse modelado formal, puede tener como resultado un modelado matemático, lingüístico, o computacional (de simulación). En cualquiera de los casos puede ocurrir que la transformación no capture toda la riqueza del modelo conceptual, y la validación lógica deberá asegurarse de que se alcanza el nivel adecuado de validez.
- Validación de aptitud: El objetivo primario de la validación de aptitud es evaluar la adecuación del modelo formal en relación con su utilidad, capacidad, representatividad y costo, desde el punto de vista de los potenciales usuarios.

2.4.3. Validación de los resultados de un modelo

1. Obtención de datos del mundo real

Para obtener un modelo válido, los analistas deben intentar medir las entradas y las salidas del sistema real, así como los atributos de variables intermedias.

En la práctica, los datos estarán disponibles en diferentes cantidades, como se muestra en las cuatro siguientes situaciones:

1. A veces es difícil o incluso imposible obtener datos relevantes. Por ejemplo, en la simulación de una guerra nuclear, es imposible tomar los datos necesarios.
2. En ocasiones, si no está disponible el sistema real hay que recurrir a algún sistema parecido. Los militares, comúnmente, realizan pruebas de campo para predecir futuras situaciones.
3. Usualmente es posible recoger sólo algunos datos correspondientes a ciertos períodos históricos o ciertas partes del sistema.
4. En otras aplicaciones, existe una sobrecarga de datos de entrada, en particular si estos datos se recogen electrónicamente.

Cuanto más retrocedan los analistas en el tiempo en busca de información, más cantidad de datos podrán recoger y más seguras serán las pruebas de validación, a menos que, vayan tan atrás que, las leyes que gobiernan el sistema sean ya diferentes, en ese caso, los datos serían inútiles.

2. Técnicas para comparar datos reales y simulados

Supongamos que los analistas han tenido éxito al obtener datos del sistema real y desean validar el modelo de simulación. Entonces se deben introducir esos datos en el modelo, en orden histórico. Tras

ejecutar el programa de simulación, se obtienen una serie de valores de salida que se compararán con los datos de salida que poseemos del sistema real.

Tras haber validado el modelo de simulación, se deberán comparar diferentes escenarios usando entradas desordenadas, no entradas históricas. De hecho, la historia nunca se va a repetir exactamente.

Para comparar una serie de valores de salida del modelo de simulación con una serie de valores de salida en orden histórico del sistema real, se hallan disponibles algunas técnicas:

2. Representar en un plano los valores de salida del sistema real y del sistema simulado, en donde el eje horizontal denota tiempo y el eje vertical denota los valores reales y simulados. A simple vista y comparando las dos series de puntos podremos decidir si el modelo de simulación refleja de forma precisa el fenómeno de interés.
3. Otra técnica simple es la prueba de Schruben-Turing¹. Los analistas muestran a sus clientes una mezcla de las series de valores del sistema real y simulado, y les invitan a identificar qué datos han sido generados por el ordenador. Naturalmente, estos clientes pueden identificar correctamente algunos datos por casualidad. No obstante, esta coincidencia se tendrá en cuenta si así se demuestra estadísticamente.
4. En lugar de inspeccionar a simple vista las series de valores reales y simulados, se pueden usar estadísticas matemáticas con objeto de obtener datos cuantitativos acerca de la calidad del modelo de simulación.

1. Análisis de sensibilidad

Los modelos y submodelos (módulos) dotados de entradas y salidas no observables no pueden estar sujetos a las pruebas anteriormente citadas. En ese caso, se aplicará análisis de sensibilidad, a fin de determinar si el comportamiento del modelo está de acuerdo con los juicios de los expertos.

El análisis de sensibilidad se define como la investigación sistemática de la reacción de la salida del modelo ante cambios drásticos en la entrada y en la estructura del modelo.

El análisis de sensibilidad debe aplicarse con el objeto de descubrir qué entradas son realmente importantes. Esta información es útil, incluso si hay muchos datos en la salida y en la entrada del sistema simulado. Es preciso esforzarse, si es posible, en recoger información en las entradas importantes.

Sin embargo, puede que sea imposible recoger información en esas entradas, entonces, los analistas deberán aplicar una técnica denominada «ajuste a distribuciones». Esta consiste en deducir la distribución de probabilidad de los valores de entrada, haciendo uso del conocimiento de personal experto. A continuación se realiza un muestreo para generar valores de entrada de esa distribución. Finalmente, esos valores son introducidos en el modelo de simulación, el cual entrega una distribución de probabilidad de valores de salida.

2.4.4. Conclusiones

No existe un criterio universal a la hora de contrastar la validez de un modelo, tanto en sus aspectos conceptuales como técnicos. Para cualquier tipo de modelo de simulación se ha constatado la importancia de considerar el propósito para el que fue construido en el proceso de validación, además de lo poco definitivas que resultan las pruebas de naturaleza estadística. Éstas tan sólo sirven para otorgar al modelo cierta confianza cuando se usa para predecir el comportamiento del sistema real, pero no al explicar su funcionamiento interno. Por ello, los modelos validados de este modo no serán provechosos si lo que pretendemos es evaluar posibles mejoras en el mundo real.

3. Lenguajes de Simulación

3.1. Introducción

Existen varias alternativas para resolver un problema de simulación:

- Utilizar una herramienta como Excel, que permite desarrollar modelos simples rápidamente.
- Utilizar un lenguaje de programación de propósito general, como FORTRAN.
- Utilizar un lenguaje de simulación, como GPSS.
- Utilizar un simulador, como SIMPROCESS.
- Utilizar un híbrido de lenguaje de simulación y simulador, como Extend.

La alternativa más rápida y accesible para la mayoría de los usuarios consiste en la utilización de Excel. Para modelos muy simples que no tendrán e simulacimayores modificaciones en el futuro, esta alternativa es válida. Cuando el modelo es más complejo, Excel deja de ser una alternativa viable ya que no provee

flexibilidad para el crecimiento y el mantenimiento de un modelo complejo en Excel es muy engorroso. Por lo tanto, para problemas complejos del mundo real, Excel no es una posible solución.

A pesar de que muchos modelos de simulación se escriben en lenguajes de propósito general, como FORTRAN, PL/I, o Pascal, se han desarrollado lenguajes de simulación específicos para asistir en la escritura de programas de simulación. Algunos de los lenguajes de simulación más populares son GPSS, SIMSCRIPT, SIMAN, y SLAM.

La principal ventaja de utilizar un lenguaje de simulación específico comparado con un lenguaje de propósito general es que se requieren menos líneas de código, lo que permite que un programa se desarrolle con mayor velocidad y facilidad. Además, muchos lenguajes de simulación incluyen posibilidades de animación, lo cual permite al usuario ver los efectos de la simulación a medida que el programa es ejecutado. Esto es especialmente valioso para identificar posibles cuellos de botella en el sistema o áreas críticas que requieren un modelado más detallado.

Una alternativa consiste en desarrollar la simulación con un programa simulador. Estos programas contienen objetos que pueden ser manipulados gráficamente para crear fácilmente un modelo de simulación. Los modelos resultantes generalmente permiten observar una animación del proceso y proveen los resultados en formato tabular o gráfico. Algunos simuladores populares actualmente son SIMPROCESS, Alpha/Sim, y ProModel.

Los primeros programas simuladores no poseían capacidades de programación, lo cual los hacía inflexibles. La tendencia actual es combinar la facilidad de uso de un simulador gráfico con la flexibilidad de un lenguaje de simulación. Un ejemplo de estos nuevos híbridos entre simuladores y lenguajes de simulación es Extend, que será analizado en detalle en el próximo capítulo.

En esta sección se verán las ventajas y desventajas de los lenguajes de simulación, luego se clasificará el software de simulación y, por último, se describirán características deseables de un lenguaje de simulación. En el Anexo B se encuentra una tabla con los principales lenguajes de simulación y algunas de sus características de acuerdo con el Simulation Software Survey de la publicación OR/MS Today de Agosto de 2003.

3.2. Ventajas y desventajas de los lenguajes de simulación

La elección del lenguaje es una de las decisiones más importantes que debe tomar el modelador. Una elección inadecuada puede llevar al fracaso del proyecto de simulación, o a completarlo demasiado tarde.

Algunas de las ventajas de elegir un lenguaje de simulación frente a uno de propósito general, como FORTRAN, C, Pascal o BASIC son:

- Los lenguajes de simulación proveen la mayoría de las características s caracterrmulacie elegir un lenguaje de simulacique se necesitan para resolver un modelo de simulación, lo cual resulta en una disminución del tiempo de programación.
- Los lenguajes de simulación dan un marco natural para el desarrollo de un modelo. Los bloques con los que se construyen programas provistos por estos lenguajes están más relacionados con la simulación que los que proveen los lenguajes de propósito general.
- Los modelos de simulación son más fáciles de modificar cuando están escritos en un lenguaje de simulación.
- Los lenguajes de simulación proveen mejor detección de errores, ya que muchos tipos de errores potenciales ya han sido identificados y son chequeados en forma automática.

Sin embargo, muchos modelos de simulación siguen siendo escritos en lenguajes de propósito general. Algunas ventajas de esa elección son las siguientes:

- La mayoría de los modeladores ya conoce un lenguaje de propósito general, pero no uno de simulación.
- FORTRAN, BASIC o C están disponibles en prácticamente cualquier computadora, pero un lenguaje de simulación en particular puede que no esté disponible para la computadora que el modelador desea utilizar.
- Un programa eficiente escrito en un lenguaje de propósito general suele requerir menor tiempo de ejecución que uno equivalente en un lenguaje de simulación. Esto se debe a que los lenguajes de simulación están diseñados para modelar una gran variedad de sistemas con un conjunto de bloques, mientras que un lenguaje de propósito general puede ser ajustado a las necesidades de un sistema en particular.
- Algunos lenguajes de propósito general proveen mayor flexibilidad que la mayoría de los lenguajes de simulación.
- El costo del software suele ser menor (pero no necesariamente el costo del proyecto).

Como se puede ver, la decisión de utilizar un lenguaje de simulación o uno de propósito general debe analizarse para cada sistema en particular, ya que cada enfoque tiene sus ventajas y desventajas.

3.3. Clasificación del software de simulación

Actualmente hay tres clases de software de simulación: lenguajes de simulación, simuladores e híbridos.

Un lenguaje de simulación es un lenguaje de programación que contiene características especiales para ciertos tipos de aplicaciones de simulación. Por ejemplo, SIMAN y SLAM II tienen módulos de manufactura para cintas transportadoras y vehículos guiados automatizados. Un modelo se desarrolla en un lenguaje de simulación escribiendo un programa que utiliza las construcciones que provee el lenguaje. La mayor ventaja de la mayoría de los lenguajes es la posibilidad de modelar casi cualquier tipo de sistema, sin importar los procedimientos o lógica de control del sistema. Posibles desventajas son la necesidad de experiencia de programación y el tiempo de codificación y depuración asociado al modelado de sistemas complejos.

Un simulador es un software que permite simular un sistema de una clase específica, sin escribir un programa. Por ejemplo, existen simuladores para ciertos tipos de sistemas de manufactura, de computación y de comunicaciones. El sistema que se necesita es seleccionado para simular mediante menús y gráficos, sin necesidad de programar. La gran ventaja de un simulador es que el desarrollo del modelo puede tomar mucho menos tiempo que en un lenguaje de simulación. Esto puede ser muy importante ya que los tiempos disponibles son muy cortos en muchos ambientes de negocios. Otra ventaja es que la mayoría de los simuladores tienen bloques para construir modelos relacionados específicamente con los componentes de la clase del sistema a modelar. Además, gente sin experiencia en programación puede desarrollar modelos utilizando simuladores debido a su facilidad de uso. La mayor desventaja de los simuladores es que están limitados a modelar sólo aquellos sistemas que se ajustan a sus características, lo cual les resta flexibilidad.

Por último, un híbrido entre un lenguaje de simulación y un simulador es un software que posee las características de un simulador, y además provee la posibilidad de agregar programación al modelo. Esto permite que gran parte del modelo sea desarrollada utilizando menús y gráficos; y las partes más complejas, para las que no existen bloques ya programados, sean desarrolladas en un lenguaje provisto por el híbrido.

3.4. Características deseables de un lenguaje de simulación

A continuación se describirán algunas características que se consideran deseables en un lenguaje de simulación. Las mismas se han agrupado en cinco categorías.

3.4.1. Características generales

La característica más importante para un software de simulación es la flexibilidad para el modelado. Si el software de simulación no posee la capacidad necesaria para modelar una aplicación en particular, el sistema constituirá una aproximación, reduciéndose su utilidad. Las entidades del modelo deben poseer atributos generales, que puedan ser modificados.

La facilidad para el desarrollo del modelo es una característica muy importante, ya que los tiempos disponibles para los proyectos suelen ser muy cortos. La exactitud y velocidad del proceso de modelado será aumentada si el software provee ayuda para el proceso de depuración, como por ejemplo un programa de depuración interactivo, control de errores, y ayuda online.

Una alta velocidad de ejecución del modelo es particularmente importante para modelos muy grandes, como por ejemplo aplicaciones militares.

El tamaño máximo permitido para el modelo puede ser un factor importante también para modelos muy grandes.

Es deseable que el software esté disponible en distintos tipos de computadoras y para distintos sistemas operativos, y que los modelos sean compatibles entre estas clases.

Por último, en algunas aplicaciones es deseable que el software posea capacidades para desarrollar simulaciones discretas-continuas combinadas.

3.4.2. Animación

Las características de animación de fácil uso son una de las razones por las cuales se incrementó la popularidad del modelado de simulación. En una animación, los elementos más importantes de un sistema se representan en un monitor mediante iconos que cambian de forma, color, o posición cuando hay un cambio de estado en la simulación. Por lo tanto, se puede ver el cambio de un sistema en el tiempo en forma gráfica. La mayoría de los paquetes de animación operan en modo concurrente, es decir que la animación se muestra mientras la simulación está siendo ejecutada. Sin embargo, otros paquetes operan en modo playback, donde la animación se muestra luego de terminada la ejecución.

La razón más importante para la popularidad de la animación es que permite comunicar la esencia de un

modelo de simulación a los usuarios, aumentando la credibilidad del modelo y facilitando su comprensión.

Otras ventajas de la animación son:

- Facilitar la depuración de un modelo.
- Mostrar que un modelo no es válido.
- Sugerir mejoras a procedimientos operacionales o lógica de control de un sistema.
- Entender el comportamiento dinámico de un sistema.
- Entrenar al personal operativo.

Se debe tener en cuenta que la animación no reemplaza a un análisis estadístico detallado de la simulación, sino que lo complementa. No se puede llegar a la conclusión de que un modelo está bien definido simplemente mirando su animación. Además, agregar animación a un modelo suele aumentar el tiempo de desarrollo. Por último, solo parte de la lógica del modelo puede ser vista durante la simulación, por lo cual una animación correcta no garantiza un modelo correcto.

3.4.3. Capacidades estadísticas

Dado que la mayoría de los sistemas del mundo real tienen algún tipo de comportamiento aleatorio, es importante que un paquete de simulación contenga capacidades estadísticas que permitan modelar este aspecto. En general, cada fuente de aleatoriedad de un sistema (tiempo entre arribos, tiempo de servicio, etc.) debe ser modelado según su distribución de probabilidad, y no sólo con su promedio. Un paquete de simulación debe incluir un conjunto de distribuciones estándar (por ejemplo exponencial, gama, triangular), debe poder utilizar distribuciones basándose en datos observados del sistema, y debe contener generadores de números aleatorios.

Dado que las entradas se generan con un componente aleatorio, las salidas también tienen un componente aleatorio y es necesario utilizar técnicas estadísticas para interpretar los resultados. Un paquete de simulación debe tener un comando que permita ejecutar replicaciones independientes del modelo en forma automática, y que cada replicación utilice números aleatorios diferentes.

3.4.4. Soporte

La mayoría de los usuarios de software de simulación requieren algún tipo de soporte permanente. El distribuidor del software debe realizar cursos y seminarios públicos en forma regular. Además, el distribuidor debe proveer soporte técnico en forma rápida para solucionar los problemas que pueda encontrar el usuario. Es importante una buena documentación, que incluya libros, manuales y numerosos ejemplos. También son útiles las pruebas y demos gratuitas para que el interesado pueda evaluar el producto antes de comprarlo.

3.4.5. Reportes

Un paquete de simulación debe proveer reportes estándar para las estadísticas de performance más comunes (por ejemplo utilización, tamaño de las colas y demoras), pero también debe facilitar el desarrollo de reportes personalizados. Es importante contar con reportes que incluyan gráficos (histogramas, gráficos de barras, etc.) y poder acceder a valores particulares que sean necesarios.

4. Extend

4.1. Introducción

Extend es una herramienta de simulación de última generación, que permite desarrollar modelos dinámicos de procesos de la vida real para una gran variedad de áreas. Con Extend, se crea un diagrama de bloques de un proceso, donde cada bloque describe una parte del proceso.

Extend fue desarrollado por la empresa Imagine That Inc., fundada en 1987 por Bob Diamond, actual presidente de la empresa.

Extend permite crear modelos rápidamente, ya que provee todos los bloques que se requieren para la mayoría de las situaciones. Estos bloques son como macros, por lo tanto se puede desarrollar un modelo completo sin escribir una sola ecuación. Además, Extend permite desarrollar bloques propios para aplicaciones especializadas que requieran de bloques no provistos en las librerías incluidas.

Las principales características de Extend son:

- Incluye un conjunto de bloques que permite desarrollar un modelo rápida y fácilmente.
- Permite realizar animaciones que mejoran las presentaciones.
- La interfaz muestra gráficamente las relaciones entre las partes del sistema que se está modelando.

- Permite descomponer en forma jerárquica los modelos complejos, para facilitar su construcción y comprensión.
- Provee cuadros de diálogo que permiten cambiar valores del modelo, para realizar pruebas rápidamente.
- Permite modificar configuraciones mientras se está ejecutando el modelo.
- Provee herramientas de optimización.
- Permite desarrollar reportes personalizados para presentaciones y análisis.
- Tiene capacidad de definir costos por actividad, para realizar análisis de costos.
- Conectividad e interactividad con otros programas y plataformas, a través de Copiar/Pegar, importar/exportar, archivos de texto, XCMDs y Apple Events (Mac OS), o DLL (dynamic link library) y DDE (dynamic data exchange) y OLE/COM/ActiveX (Windows).

Además, el ambiente de Extend provee características avanzadas que lo convierten en uno de los sistemas de simulación más avanzados disponible para equipos de escritorio:

- Simulación multipropósito: Extend permite modelar sistemas de distintos dominios dinámicamente, ya sean continuos, discretos, lineales, no lineales e híbridos.
- Lenguaje de programación incluido: Se pueden modificar los bloques de Extend, o crear bloques nuevos, mediante un lenguaje de programación compilado, similar al C.
- Soporte para scripts: Se puede crear y modificar modelos remotamente, mediante un bloque de Extend o desde otra aplicación.
- Soporte integrado para otros lenguajes: Extend provee una API que permite acceder a código creado en Delphi, C++ Builder, Visual Basic, Visual C++, etc.
- Basado en librerías: Los bloques que se crean para construir modelos se almacenan en librerías, y pueden ser fácilmente reutilizados.
- Más de 600 funciones: Se pueden acceder directamente funciones para integración, estadísticas, colas, animaciones, matemáticas, matrices, sonidos, vectores, FFT, depuración, XCMDs, DLLs, manipulación de strings y bits, entrada/salida, etc. También se pueden definir funciones propias nuevas.
- Envío de mensajes: Los bloques pueden enviar mensajes a otros bloques en forma interactiva.
- Capacidad sofisticada de paso de datos: Se pueden pasar valores, vectores, o estructuras compuestas de vectores.
- Modelos de gran tamaño: El tamaño de los modelos solo está limitado por las capacidades del sistema donde se ejecuta.
- Compatibilidad multiplataforma: Se pueden correr modelos en sistemas Windows o Mac OS. Los modelos y librerías son compatibles entre estas plataformas, lo que permite que se trabaje en forma cooperativa entre plataformas diferentes.

Un modelo de Extend es un documento que contiene componentes (llamados bloques) interconectados. Cada bloque contiene información de procedimientos y datos ingresados por el usuario. Una vez creado un modelo, puede ser modificado agregando o quitando bloques, moviendo conexiones, y cambiando datos de los bloques.

Extend puede usarse en varios niveles:

- Ejecutar modelos ya desarrollados para explorar alternativas modificando datos de entrada. Si se trabaja en equipo, un usuario puede ser el que crea los modelos, mientras que otros se encargan de ejecutarlos para experimentar. Existen versiones especiales (RunTime y Player) que permiten ejecutar modelos creados por otros.
- Crear modelos utilizando los bloques provistos por Extend. El producto provee un conjunto de librerías de bloques que permite desarrollar modelos tanto continuos como discretos. Para desarrollar el modelo, se arrastran los bloques de los menús de librerías y se interconectan.
- Crear nuevos bloques usando el ambiente de programación de Extend. También se pueden modificar los bloques provistos por Extend.
- Automatizar el desarrollo de modelos utilizando scripts o automatización OLE/COM desde otras aplicaciones.

En esta sección se describirán las versiones de Extend, luego el ambiente de trabajo, las librerías, las funciones de entrada y salida que provee, el lenguaje de programación y, por último, las opciones de soporte. En el Anexo C se describen las características avanzadas de Extend según la versión a partir de la cual están disponibles, y en el Anexo D se detallan las características del lenguaje de programación ModL.

4.2. Versiones

Existen varias versiones de Extend. Se diferencian principalmente en las librerías que incluyen. A continuación se describen las diferentes versiones.

4.2.1. Extend CP (Continuous Process)

Extend CP es la base de la familia de herramientas de simulación Extend. Extend CP permite modelar sistemas o procesos que pueden representarse mediante un flujo de tiempo o valores continuo.

Este paquete incluye las librerías *Generic* y *Electronics* para simular sistemas continuos. El ambiente de desarrollo permite crear los componentes personalizados que se necesiten.

Otras características de Extend CP son:

- Modelado jerárquico a cualquier nivel.
- Navegador para explorar la estructura del modelo.
- Animaciones 2D personalizadas con niveles, colores, texto, etc.
- *Evolutionary Optimizer* incluido para encontrar soluciones precisas rápidamente.
- Reportes personalizados con conexión a MS Office.
- Ambiente de desarrollo para crear interfaces propias.
- Cuadros de diálogo, controles y alertas interactivos.
- Controles ActiveX y objetos COM embebidos.
- Conexión con bases de datos mediante ODBC/SQL.

4.2.2. Extend OR (Operations Research)

Extend OR es la herramienta esencial para simular la operación de cualquier sistema o proceso. Agrega una arquitectura de eventos discretos a Extend CP. Extend OR permite modelar sistemas o procesos que involucran objetos físicos o lógicos moviéndose en un sistema en forma discreta.

Extend OR contiene todas las características de Extend CP, más módulos y características para modelar sistemas discretos:

- Atributos personalizados y direccionamiento basado en reglas.
- Análisis estadísticos con intervalos de confianza.
- Activity Based Costing integrado para análisis financieros.
- Prioridades, retractos y bloqueos.
- Períodos de preparación (warm-up) para análisis estadísticos.
- Algoritmos de cola robustos².
- Control de recursos, decisiones y actividades.
- Repositorios de recursos para optimizar restricciones.
- Diagramas de Gantt para monitorear y analizar el estado de los componentes del sistema.

4.2.3. Extend Industry (Industrial Strength)

Diseñado para cumplir con las necesidades de una gran variedad de industrias, Extend Industry es el paquete para simular proyectos de gran escala.

Extend Industry agrega a Extend OR una base de datos relacional interna para administrar datos en forma centralizada, un módulo de flujo y herramientas adicionales para simulaciones de gran tamaño. Este paquete contiene todas las funcionalidades de Extend CP y Extend OR, más:

- Base de datos relacional para crear, ver y manipular información según tipo de producto, ubicación, componente o cualquier otra característica común.
- Separación de los datos del modelo, para mejor administración del proyecto.
- Tablas de la base de datos para proveer salidas y reportes en el modelo.
- Un Add-in de MS Excel para crear workbooks y front-ends.
- Interfase integrada con MS Access para rápido intercambio de datos.
- Diagramas de Gantt en 3D para ver el estado de los procesos dinámicamente.

4.2.4. Extend Suite (Total Simulation Project Support)

Extend Suite es un conjunto de herramientas integradas completo. Un software de análisis de última generación con el paquete Extend Industry. Extend Suite se utiliza cuando se requiere administración de datos intensiva, ajustes a distribuciones de probabilidad, o animaciones a nivel profesional.

Incluye todas las características de Extend CP, Extend OR y Extend Industry. Además:

- Stat::Fit de Geer Mountain Software determina qué distribución de probabilidad se ajusta más a los datos del mundo real provistos. La función Auto::Fit automáticamente ajusta los datos a distribuciones continuas o discretas, y provee comparaciones relativas entre las distribuciones. Las distribuciones resultantes pueden ser exportadas directamente a Extend.
- Proof Animation de Wolverine Software permite crear animaciones avanzadas para presentaciones a los usuarios. Permite crear animaciones vectoriales, que muestran distribuciones físicas y recorridos. Se puede controlar el paso de la animación e investigar comportamientos inusuales mediante aceleracio-

nes, retrasos, o saltos. Se puede realizar zoom en cualquier momento a cualquier nivel. Permite importar objetos CAD de archivos dxf. Proof Animation usa DirectX para mostrar animaciones de gran número de objetos y provee las características necesarias para realizar animaciones a nivel profesional.

4.2.5. Tipos de Uso y Costos del Software

	Extend Suite \$3,995.00	Extend Industry \$2,295.00	Extend OR \$1,595.00	Extend CP \$895.00
Ingeniería/Científico	X	X	X	X
Biológico				
Químico				
Procesos Continuos				
Sistemas de Control				
Ingeniería Electrónica				
Ambiental				
Finanzas y Economía				
Ingeniería Mecánica				
Medicina y Farmacología				
Matemáticas y Física				
Negocios/Manufactura	X	X	X	
Sistemas Bancarios				
Call Centers				
Sistemas de Comunicaciones				
Sistemas de Eventos Discretos				
Sistemas de Salud				
Sistemas de Seguros y Financieros				
Dinámica de Inventarios y Supply Chain				
Producción, Manufactura				
Logística				
Sistemas de Servicios y Ventas				
Transporte y Manejo de Materiales				
Gran Escala	X	X		
Transacciones de ATM				
Automotriz				
Plantas Embotelladoras				
Call Centers de gran volumen				
Procesos Químicos				
Logística de Distribución				
Flujo de Fábricas				
Procesamiento de Comida				
Redes				
Almacenamiento				
Análisis de Datos	X			
Ajustar datos a mejor distribución				
Selección de distribución sin datos				
Presentaciones profesionales				
Importar objetos CAD				

Figura 4.1: Tipos de uso y costos¹ para versiones de Extend

4.3. Ambiente de Trabajo

La mayor parte del trabajo de simulación en Extend se realiza con modelos. La Figura 4.2 muestra la estructura de un modelo de Extend. Los bloques se arrastran desde librerías hacia el modelo, luego se llenan los cuadros de diálogo de los bloques y, en caso de ser necesario, se relacionan con aplicaciones externas.

En dicha figura se observa el ambiente de trabajo de Extend. El modelo, Lake Pollution en este ejemplo, se encuentra en la parte superior izquierda. En la parte inferior izquierda se muestra el cuadro de diálogo correspondiente al bloque *Holding Tank* . A la derecha se encuentran las librerías utilizadas, *Utilities* , *Plotter* y *Generic* .

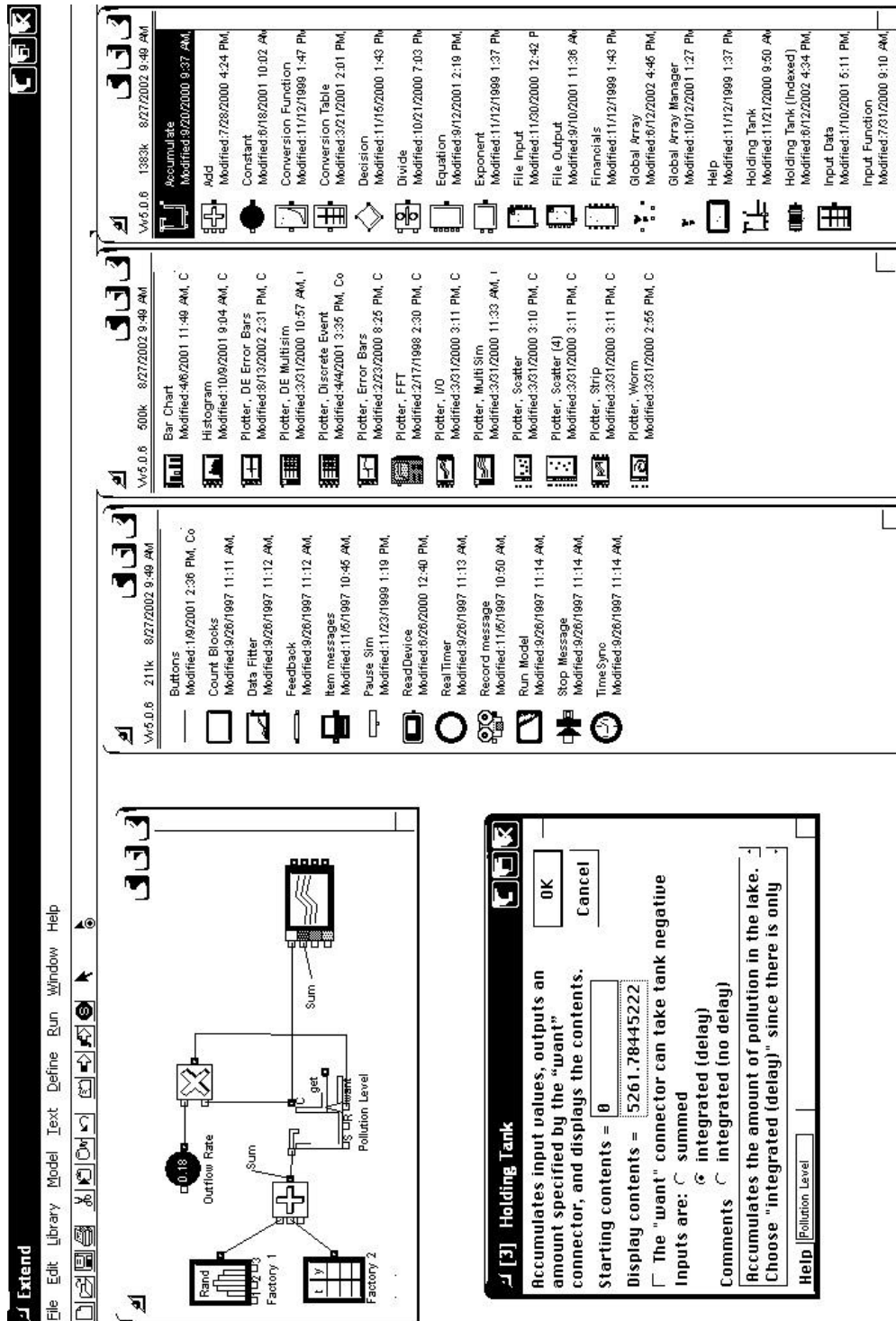


Figura 4.2: Ambiente de Trabajo en Extend

Los bloques proveen comportamiento (o código), ayuda, iconos, cuadros de diálogo, y datos predeterminados para cada paso en un proceso. Las partes más importantes del modelo son los bloques, las librerías donde se almacenan los bloques, los cuadros de diálogo asociados a cada bloque, los conectores en cada bloque, y las conexiones entre los bloques.

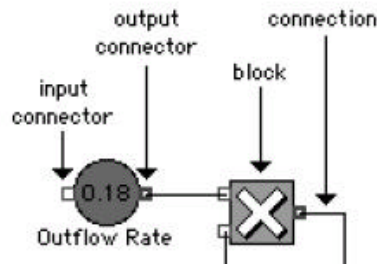


Figura 4.3: Partes de un modelo

4.3.1. Bloques

La mayoría de los bloques en Extend están compuestos por un icono, un cuadro de diálogo para ingresar datos, y conexiones. Un bloque especifica una acción o proceso. Al bloque ingresa información, el programa incluido en ese bloque la procesa, y luego la transmite al siguiente bloque en la simulación.

Cualquier bloque puede crear, modificar, o presentar información, y muchos bloques realizan más de una de estas funciones. Algunos sólo presentan información que pasa a otros bloques. Otros modifican información que pasa a través de ellos. Algunos, llamados jerárquicos, agrupan conjuntos de otros bloques.

4.3.2. Librerías

Las librerías son repositorios de bloques. La definición de un bloque (su programa, icono, cuadro de diálogo) se almacena en la librería. Cuando se incluye un bloque en un modelo, el bloque no se copia al modelo, sino que se referencia a la librería. Extend almacena los datos ingresados en los cuadros de diálogo en el modelo.

Hay varias ventajas en referenciar los bloques y no copiarlos. Si se cambia la definición de un bloque en una librería, se actualizan automáticamente todas las instancias del bloque. También, las definiciones de los bloques son muy grandes, almacenar sólo referencias ahorra gran cantidad de espacio en memoria.

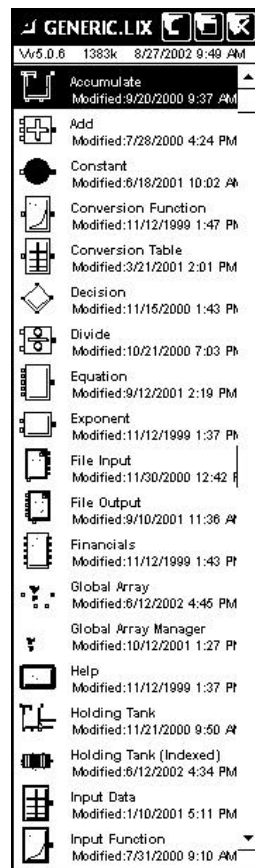


Figura 4.4: Librería *Generic*

4.3.3. Cuadros de diálogo

La mayoría de los bloques tiene un cuadro de diálogo asociado a ellos. Los cuadros de diálogo se usan para ingresar valores y configuraciones antes de ejecutar la simulación, y ver los resultados mientras se ejecuta la simulación. Los cuadros de diálogo se abren haciendo doble click en el bloque elegido.

Se puede encontrar más información acerca de un bloque haciendo click en la opción *Help* del cuadro de diálogo, que abre la ventana de ayuda correspondiente a ese bloque. Los cuadros de diálogo pueden dejarse abiertos mientras se ejecuta la simulación, lo cual permite ver los valores que reporta el bloque durante la ejecución. Inclusive se pueden cambiar los valores y configuraciones durante la ejecución.

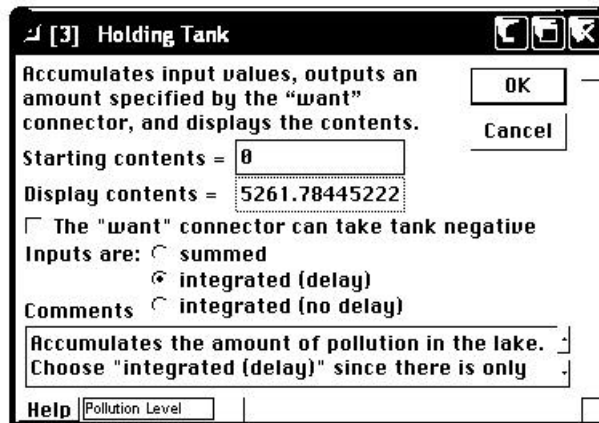


Figura 4.5: Cuadro de diálogo para un *Holding Tank*

4.3.4. Conectores y conexiones

La mayoría de los bloques en Extend tienen conectores de entrada y salida. La información ingresa en un bloque a través de los conectores de entrada, y egresa por los conectores de salida. Un bloque puede tener varios conectores de entrada y salida, como también puede no tener ninguno.

Se utilizan líneas para conectar bloques. Estas líneas, llamadas conexiones, muestran el flujo de información entre bloques a través del modelo. La simulación es una serie de cálculos y acciones que tienen lugar a través de las conexiones en forma repetitiva. Cada repetición se denomina *step* en un modelo continuo y *evento* en un modelo discreto.

Existen dos tipos de conexiones, de ítems (*Item*) y de valores (*Value*). Las conexiones de ítems representan objetos que se mueven a través del sistema. Los ítems pueden tener atributos y prioridades asociados. Ejemplos de ítems son partes, pacientes, y paquetes de información. Las conexiones de valores representan sólo un número, que varía durante la ejecución de la simulación. Ejemplo de valores son el número de ítems en una cola y el nivel de fluido en un tanque. La forma de diferenciar las conexiones es por el icono de los conectores.

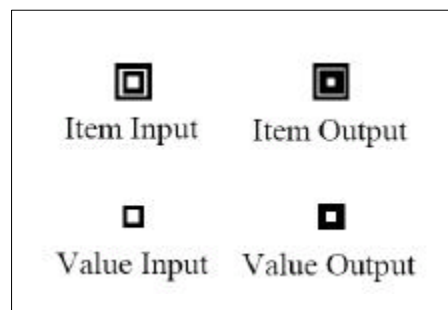


Figura 4.6: Conectores *Value* e *Item*

4.3.5. Menús y barras de herramientas

Dentro de Extend, se accede a muchas opciones desde menús y barras de herramientas. A continuación se describen las opciones más importantes del menú y barra de herramientas de Extend.



Figura 4.7: Menú y barra de herramientas

El primer grupo de iconos de la barra de herramientas es el correspondiente a *File*. Dentro del mismo se encuentran las opciones típicas de un menú de archivos, *Crear nuevo modelo*, *Abrir modelo existente*, *Guardar modelo* e *Imprimir*.

El segundo grupo es el correspondiente a *Edit*. Contiene las opciones típicas de un menú de edición, *Cortar*, *Copiar*, *Pegar* y *Deshacer*.

El tercer grupo es el correspondiente a *Model*. Las opciones que contiene son *Notebook*, *Ejecutar simulación*, *Ejecutar simulación con optimización* y *Stop*. *Notebook* permite acceder en forma rápida a parámetros del modelo que se seleccionan. *Ejecutar simulación* realiza la ejecución, mostrando una barra de estado de la ejecución y los Plotters del modelo. *Ejecutar simulación con optimización* ejecuta la simulación cuando existe un bloque de optimización. Por último, *Stop* detiene la ejecución en curso.

El cuarto grupo consiste en iconos de selección y dibujo, que facilitan el manejo del entorno.

El Menú de Extend, además de incluir las opciones de los primeros tres grupos de la barra de herramientas, contiene opciones más avanzadas. Entre ellas se destacan el submenú *Library*, que permite abrir librerías existentes, crear nuevas y cerrar librerías abiertas; *Define*, que permite crear nuevos bloques y modificar el código de bloques existentes; y *Run*, que permite configurar la ejecución del modelo.

4.4. Librerías

Las librerías son repositorios de bloques, donde están almacenadas sus definiciones. Las distintas versiones de Extend proveen distintas librerías, además es posible desarrollar librerías propias o adquirir librerías especiales desarrolladas por terceras empresas.

4.4.1. Librerías Internas

La versión básica de Extend, Extend CP, incluye librerías para modelar sistemas continuos. Las dos librerías más importantes de este grupo son *Generic* y *Plotter*. *Generic* incluye bloques que permiten realizar las operaciones más comunes de encaminamiento, almacenamiento de valores, entrada y salida, matemáticas, optimización y estadísticas. *Plotter* contiene los bloques utilizados para la impresión de resultados. Algunos de los bloques de esta librería son específicos para modelos continuos o discretos, mientras que otros pueden usarse en cualquiera de los dos tipos.

Extend además incluye las siguientes librerías:

- *Animation*: Permite agregar animaciones personalizadas a los modelos.
- *Electronics*: Incluye bloques para simular diseño de sistemas analógicos, digitales, de control y de procesamiento de señales.
- *Utilities*: Contiene una colección de bloques que permiten contar el número de bloques de un modelo, adaptar datos a una curva, sincronizar el modelo a tiempo real, medir la duración de una ejecución de la simulación, agregar botones al modelo, etc.
- Librerías de ejemplo para ilustrar las características de Extend, como por ejemplo *ModL Tips* y *Custom Blocks*.

Además de las librerías incluidas en Extend CP, los demás paquetes de Extend contienen librerías adicionales:

- Extend OR: Incluye las librerías *Discrete Event*, *Manufacturing* y *BPR*, que permiten modelar sistemas discretos.
- Extend Industry: Incluye las capacidades de Extend OR, y además librerías y características que facilitan la administración de datos y operaciones de gran volumen.
- Extend Suite: Agrega capacidades de animación profesional y ajustes a distribuciones de probabilidad a Extend Industry.

4.4.2. Librerías Externas

Existen librerías desarrolladas por terceras personas o empresas para propósitos específicos. A continuación se describen las más importantes disponibles actualmente:

- LightStone: Desarrollado por el Profesor Edward Voigtman de la University of Massachusetts para el

campo de la Química Analítica. Es un conjunto de tres librerías que permite modelar el comportamiento cuantitativo de procesos químicos. Las librerías contienen más de 300 bloques.

- The BioQUEST Library VI: Desarrollada por el BioQUEST Curriculum Consortium de Beloit College para el campo de la Biología. La librería permite desarrollar simulaciones en un conjunto de investigaciones biológicas.
- IDEAS: Desarrollado por IDEAS Simulation, Inc. para el campo del Control de Procesos Químicos. Facilita el desarrollo de simulaciones dinámicas para administrar decisiones de inversiones, procesos de diseño, prueba de sistemas de control, entrenamiento de operadores y optimización de operaciones.
- THINK-POWERS: Desarrollado por Microfusion Engineering Laboratories, Inc. Facilita el modelado de los equipos encontrados en plantas de procesamiento.
- OpEMCSS: Desarrollado por John R. Clymer and Associates para el campo de Ingeniería de Sistemas Complejos. Provee la capacidad de examinar elementos importantes en el espacio del problema, y formular soluciones alternativas. Contiene un sistema experto.
- Plant Builder: Desarrollado por Simulation Dynamics, Inc. para la construcción de plantas. Plant Builder permite modelar fácilmente desde plantas pequeñas hasta plantas sumamente complejas.
- PaperMac AB: Consiste en cuatro grupos de librerías para el Procesamiento de Pulpa y Papel. Son FlowMac, MillMac, PaperMac y PulpMac. Permiten simular los procesos relacionados con el procesamiento de pulpa y papel.
- Semiconductor & AMHS: Desarrollado por ACADZ, Inc. Ayuda a empresas de electrónica y desarrollo de semiconductores a modelar sus procesos de manufactura, que son extremadamente complejos.
- Supply Chain Builder: Desarrollado por Simulation Dynamics, Inc. para el área de Logística. Permite modelar sistemas complejos, para estudiar el impacto de cambios en la demanda, logística, y políticas de producción del sistema.

4.4.3. Uso de Librerías

Las librerías son repositorios de bloques fáciles de acceder y utilizar.

Se pueden crear bloques jerárquicos que contienen un conjunto de bloques de un modelo. Un bloque jerárquico puede almacenarse en una librería, para tener acceso a él en forma rápida si se lo utiliza frecuentemente.

La creación de una nueva librería permite el almacenamiento de nuevos bloques, ya sean programados o jerárquicos. Los bloques programados pueden ser tanto bloques nuevos creados por el usuario, o bloques existentes en Extend modificados por el usuario.

Cuando se abre un modelo, Extend abre automáticamente las librerías utilizadas por el mismo. Otras librerías pueden ser abiertas manualmente desde el menú de Extend.

La compilación de librerías o bloques es necesaria cuando se programan bloques. La compilación se realiza desde el menú de Extend. Existe la opción de proteger el código de las librerías desarrolladas por el usuario, si se desea evitar que se acceda al mismo. Una librería protegida funciona en forma similar a cualquier librería, excepto que el código de los bloques no puede ser visto ni modificado.

4.5. Entrada y Salida

Extend permite el manejo de información de entrada y salida de muchas formas. Las más importantes son:

- Plotters.
- Notebook del modelo.
- Impresión.
- Copiar/Pegar, Drag & Drop.
- Datos compartidos dentro del modelo usando *Global Array Manager*.
- Acceso a datos mediante archivos locales o remotos.
- Acceso a datos mediante bloques de comunicación a bases de datos ODBC/SQL.
- Comunicación entre procesos (IPC, Interprocess Communication).
- Objetos OLE/COM y controles ActiveX embebidos.
- Reportes.

A continuación se describirán cada uno de ellas.

4.5.1. Plotters

Extend provee un conjunto de Plotters que permiten graficar información de distintas formas. Vienen incluidos en la librería llamada *Plotter*. La mayoría de ellos tienen un conjunto de características en común.

La elección del tipo de Plotter a utilizar depende del tipo de modelo que se esté desarrollando (discreto

o continuo) y de la forma que se desea mostrar la información.

Se puede tener más de un Plotter en un modelo, y estos pueden ubicarse en cualquier lugar del modelo. Generalmente la ventana de cada Plotter se abre automáticamente cuando se ejecuta una simulación, pero esto puede configurarse para cada Plotter del modelo. Cuando se abre la ventana de un Plotter se ve el resultado de la última ejecución.

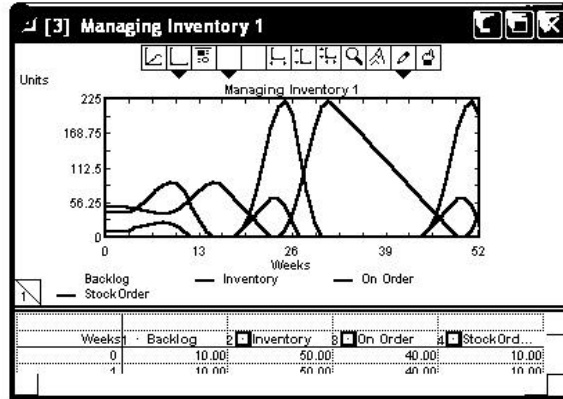


Figura 4.8: Ventana de un Plotter de tipo *Plotter*, *I/O*

La ventana de un Plotter tiene dos paneles, el superior donde se muestra el gráfico, y el inferior donde se muestran los datos del gráfico en una tabla. En el panel superior se encuentra la barra de herramientas, que permite modificar la configuración del gráfico. A través de ellas se pueden cambiar las vistas, modificar los títulos, cambiar el formato de los datos, y controlar como y cuando se muestran los datos.

Existen varios tipos de Plotters, algunos de los cuales pueden usarse en modelos continuos, otros en modelos discretos y otros en ambos.

4.5.2. Notebook del modelo

En la mayoría de los programas, los parámetros de los bloques se acceden sólo desde los cuadros de diálogo. Esto da la ventaja de que siempre se sabe donde encontrarlos. En modelos grandes, tener todas las opciones en cuadros de diálogo puede ser una desventaja. Por ejemplo, puede necesitarse acceso rápido a algunos parámetros en varios bloques que están en distintas partes del modelo. Este problema es solucionado mediante la clonación de parámetros y el uso del Notebook del modelo.

Cada modelo tiene un Notebook asociado, que puede ser utilizado para controlar parámetros del modelo, reportar resultados de la simulación, y documentar el modelo. Si se desea acceder a algunos ítems de los cuadros de diálogo en forma centralizada, se pueden clonar al Notebook, y ser accedidos desde allí. La clonación consiste en la creación de un enlace al parámetro original. En la Figura 4.9 se puede ver que hay dos parámetros de dos cuadros de diálogo independientes clonados en el Notebook. Esto hace que cualquier modificación que se realice tanto en el Notebook como en el cuadro de diálogo se vea reflejada en ambas partes.

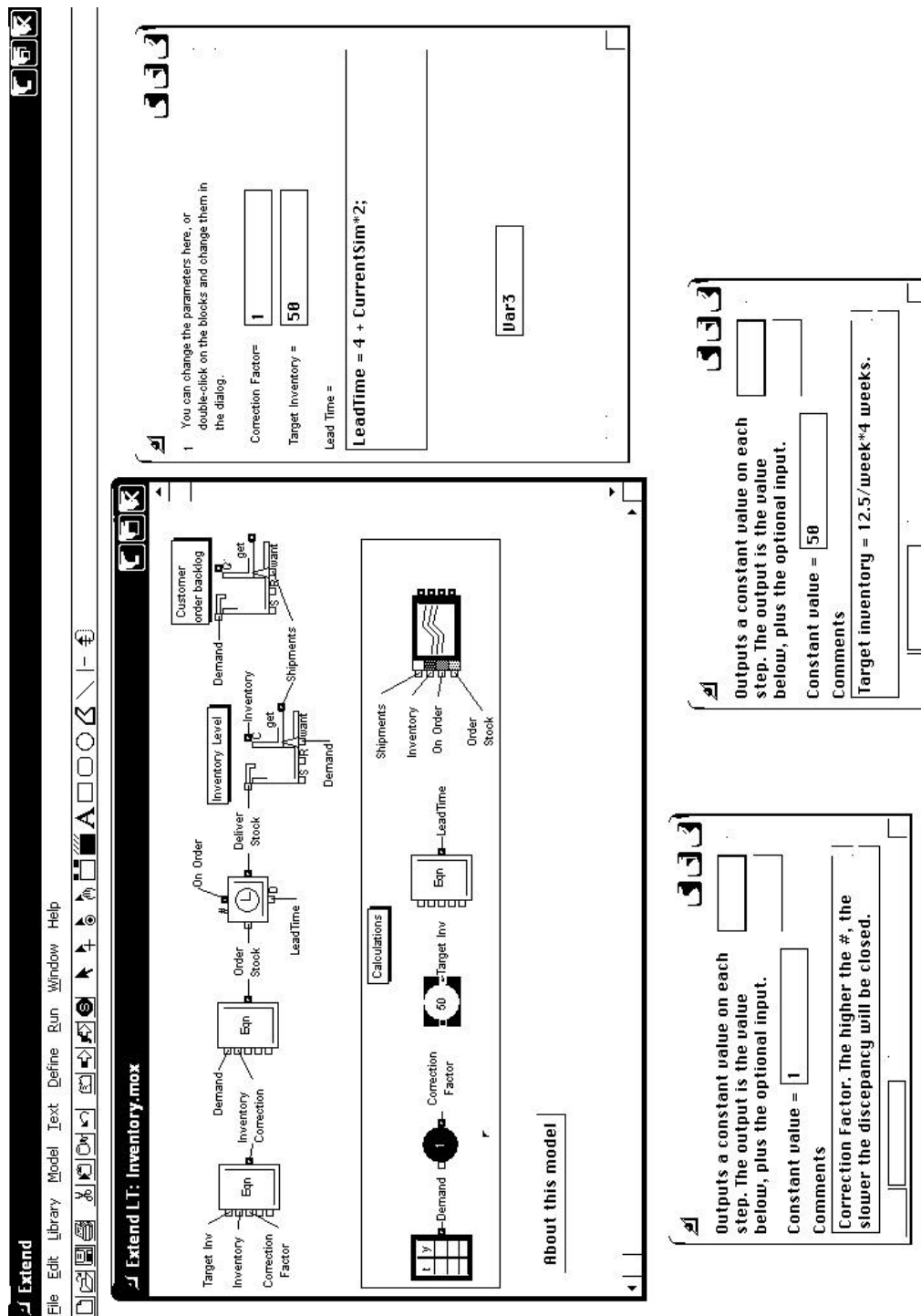


Figura 4.9: Notebook

En definitiva, un Notebook es una ventana que puede personalizarse para organizar los datos de un modelo. Puede ser usado como front-end del modelo, controlando parámetros, reportando resultados y documentando el modelo.

4.5.3. Imprimir

La forma más común de exportar datos consiste en imprimirlos en papel. Extend provee muchas opciones de impresión.

La primera elección que se debe hacer es qué se va a imprimir. Muchos programas sólo permiten imprimir el documento activo, Extend, en cambio, permite las siguientes opciones:

- Hoja de trabajo del modelo: Imprime el modelo como se ve en pantalla. Opcionalmente se pueden imprimir el contenido de los cuadros de diálogo y plotters.
- Cuadro de diálogo: Imprime el cuadro de diálogo. Se puede especificar si se imprimen o no los datos de las tablas, y si se imprime sólo la ventana principal o todas.
- Plotter: El gráfico y, en forma opcional, la tabla de datos.
- Notebook: El contenido del Notebook como aparece en pantalla.
- Texto de ayuda: El texto de la ayuda elegida.
- Archivo de texto: El texto del archivo.
- Ventana de librería: La imagen de la ventana de la librería elegida.
- Ventana de estructura: Imprime los datos de la estructura del bloque que se seleccionen (icono, conectores, cuadro de diálogo, código, ayuda).

Para imprimir, se debe seleccionar la opción *Print* del menú *File*. Extend provee opciones acerca de qué tipo de impresión se realizará según el tipo de ventana activa.

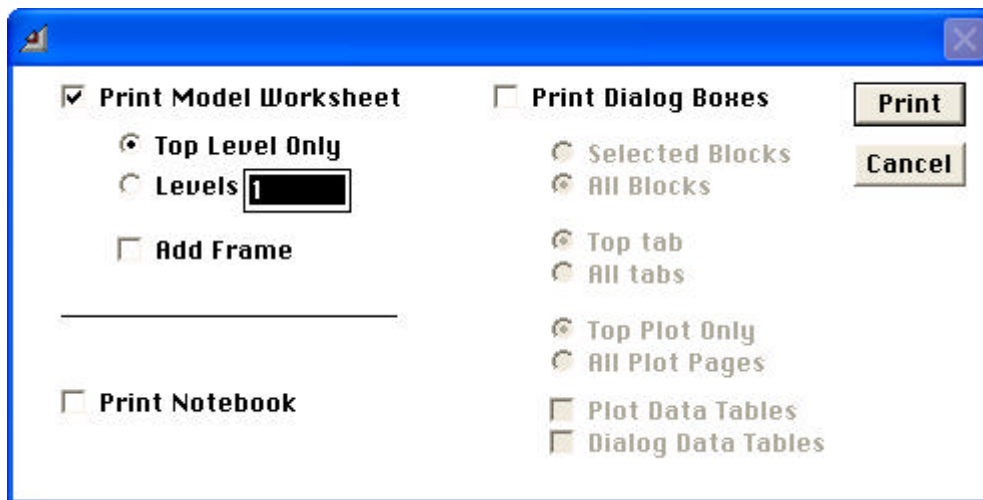


Figura 4.10: Opciones de impresión cuando la ventana activa es la del modelo

4.5.4. Copiar y Pegar, Drag & Drop

Extend utiliza las opciones de Copiar y Pegar como cualquier otra aplicación. La información se pasa en formato de texto o gráfico. Extend permite ver el contenido del portapapeles mediante la opción *Show Clipboard*.

Las opciones de Copiar y Pegar pueden ser usadas tanto dentro de Extend, como entre Extend y otras aplicaciones.

Dentro de Extend se pueden copiar bloques, en cuyo caso el portapapeles almacena todos los parámetros y conexiones. Esto permite copiar porciones de un modelo, incluyendo los valores de los cuadros de diálogo.

Extend permite copiar en forma de texto, tablas o gráficos, según el tipo de datos, a otras aplicaciones. A su vez se pueden copiar datos de otras aplicaciones hacia Extend.

Drag & Drop es la forma más fácil de mover una selección una distancia corta. Consiste en seleccionar lo que se desea mover, arrastrarlo, y soltarlo en el lugar deseado.

4.5.5. Datos compartidos dentro del modelo usando Global Array Manager

Global Arrays son vectores de dos dimensiones que pueden ser accedidos por cualquier bloque en un modelo de Extend. Los bloques de Global Arrays son *Global Array* y *Global Array Manager* de la librería *Generic*.

Estos bloques pueden usarse para:

- Compartir información entre bloques, cuando una conexión directa entre ellos no es conveniente.
- Guardar información que pueda ser accedida mediante índices de fila y columna.
- Acceder a hojas de cálculo existentes.
- Acceder a bases de datos usando ODBC/SQL.

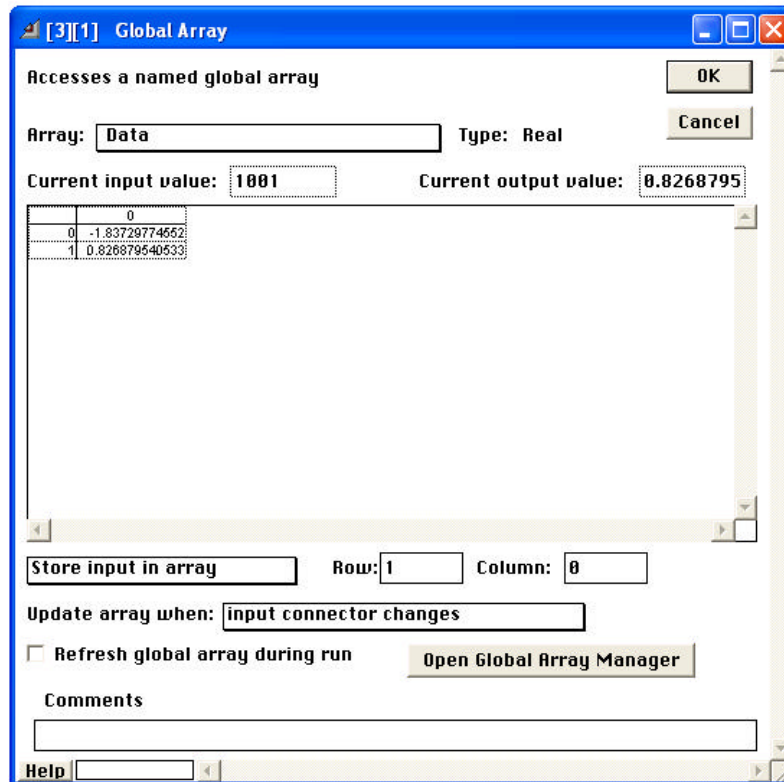


Figura 4.11: Cuadro de diálogo de un *Global Array*

4.5.6. Acceso a datos mediante archivos locales o remotos

Extend permite importar y exportar datos desde y hacia archivos de texto. Los archivos de texto pueden contener texto sin formato y datos numéricos separados por algún delimitador.

Los archivos de texto pueden estar almacenados localmente, accederse en forma remota a través de una red, o a través de Internet usando FTP.

Hay varias situaciones donde se puede necesitar leer o escribir archivos de texto:

- Cuando se desea compartir datos con una base de datos u otros programas. Prácticamente cualquier programa puede leer archivos de texto.
- Cuando se utilizan grandes cantidades de información, es más rápido usar archivos de texto que copiar y pegar.
- Las grandes computadoras, como mainframes, suelen transmitir la información en archivos de texto. Existen varias formas de utilizar archivos de texto en Extend:
- El modo más fácil de manejar archivos de texto desde bloques es usando los bloques *File Input* y *File Output* de la librería *Generic*.
- Se pueden utilizar los comandos *Import Data* y *Export Data* del menú *File*, que permiten leer y escribir archivos de texto para tablas de cuadros de diálogo y Plotters.
- Si se crean bloques propios, se pueden utilizar las funciones de entrada/salida del lenguaje para leer y escribir archivos de texto.
- El comando de Reportes, explicado más adelante en esta sección, genera archivos de texto.

4.5.7. Acceso a datos mediante bloques de comunicación a bases de datos ODBC/SQL

Se pueden acceder bases de datos mediante los bloques *Global Array Manager* usando ODBC (Open Database Connectivity).

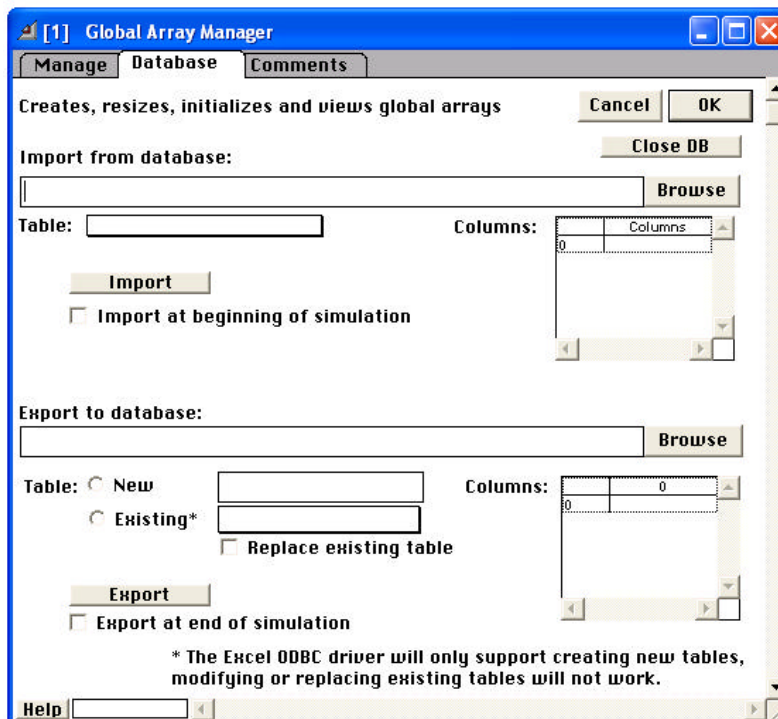


Figura 4.12: Cuadro de diálogo de un *Global Array Manager* para acceder a una base de datos

4.5.8. Comunicación entre procesos

Los bloques IPC de la librería *Generic* permiten que aplicaciones de hojas de cálculo sean parte activa de un modelo. Los bloques IPC pueden enviar y recibir datos de aplicaciones de hojas de cálculo (como Excel). Por ejemplo, pueden enviar un valor a una celda específica, para luego recibir un valor recalculado de otra celda. Los bloques *Data Send*, *Data Receive* y *SpreadsheetCalc* permiten realizar estas operaciones. Además, el bloque *Command* permite ejecutar un comando o macro en la hoja de cálculo.

4.5.9. Objetos OLE/COM y controles ActiveX embebidos

Objetos embebidos son componentes externos que permiten que una aplicación tenga nuevos comportamientos o funcionalidades sin necesidad de escribir código personalizado para el comportamiento. Extend soporta esta funcionalidad bajo Windows, haciendo fácil que los usuarios puedan incorporar objetos y controles para hacer cosas que hubieran sido más difíciles o imposibles de hacer de otra manera.

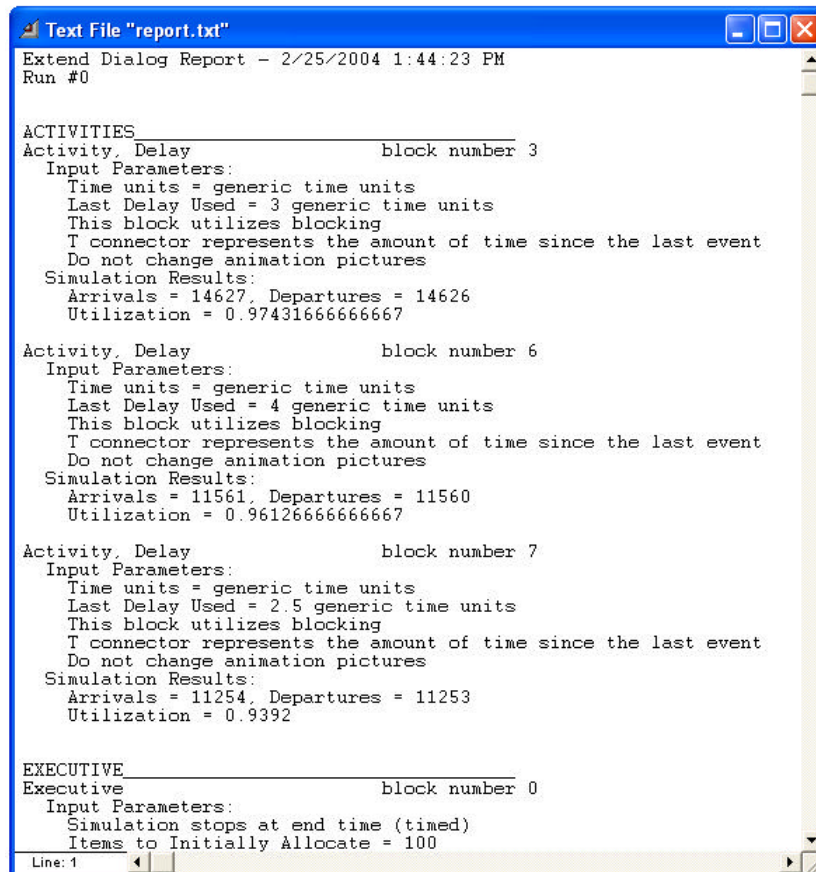
Hay dos lugares donde se pueden embeber estos objetos. El primero es en el modelo, donde pueden incluirse como objetos contenedores, y manipularse en forma similar a los bloques. El segundo es en el cuadro de diálogo del bloque, donde se pueden insertar como un tipo especial de ítem de diálogo.

La forma de embeber un objeto es a través del comando *Insert Object* del menú *File*.

4.5.10. Reportes

Los comandos de Reportes del menú *Run* permiten generar reportes personalizados de los datos del modelo. Se pueden elegir los bloques para los cuales se desean generar reportes. El reporte se genera en un archivo de texto, por lo que puede verse dentro de Extend o desde otras aplicaciones para análisis o presentación.

Hay dos tipos de reportes, de diálogos o de estadísticas. Los Reportes de diálogos incluyen los valores finales de los parámetros de entrada y salida de cada bloque seleccionado, además de la información escrita en el campo comentarios de cada bloque. Los Reportes de estadísticas incluyen los valores finales de los parámetros de salida solamente. Los resultados en los Reportes de estadísticas están ordenados en forma tabular, lo que facilita su comparación.



```
Text File "report.txt"
Extend Dialog Report - 2/25/2004 1:44:23 PM
Run #0

ACTIVITIES
-----
Activity, Delay                block number 3
  Input Parameters:
    Time units = generic time units
    Last Delay Used = 3 generic time units
    This block utilizes blocking
    T connector represents the amount of time since the last event
    Do not change animation pictures
  Simulation Results:
    Arrivals = 14627, Departures = 14626
    Utilization = 0.974316666666667

Activity, Delay                block number 6
  Input Parameters:
    Time units = generic time units
    Last Delay Used = 4 generic time units
    This block utilizes blocking
    T connector represents the amount of time since the last event
    Do not change animation pictures
  Simulation Results:
    Arrivals = 11561, Departures = 11560
    Utilization = 0.961266666666667

Activity, Delay                block number 7
  Input Parameters:
    Time units = generic time units
    Last Delay Used = 2.5 generic time units
    This block utilizes blocking
    T connector represents the amount of time since the last event
    Do not change animation pictures
  Simulation Results:
    Arrivals = 11254, Departures = 11253
    Utilization = 0.9392

EXECUTIVE
-----
Executive                      block number 0
  Input Parameters:
    Simulation stops at end time (timed)
    Items to Initially Allocate = 100
Line: 1
```

Figura 4.13: Ejemplo de Reporte de diálogos

4.6. Lenguaje de Programación

Extend provee un lenguaje de programación llamado ModL, que permite crear nuevos bloques o modificar bloques existentes. Esto hace que las posibilidades de Extend no estén limitadas a las funcionalidades incluidas en las librerías de bloques, sino que es posible desarrollar nuevos bloques que realicen tareas específicas para el modelo que se está creando. A continuación se verán como están compuestos los bloques y los pasos para crear código ModL. En el apéndice D se encuentra una descripción detallada del lenguaje ModL.

4.6.1. Partes de un bloque

Los bloques constan de seis partes:

Cuadro de diálogo	El cuadro de diálogo que se ve al hacer doble click en el icono del bloque. Se especifican todos los botones, textos, y campos de entrada que forman el cuadro de diálogo. Se pueden separar los ítems de los cuadros de diálogo en distintas lengüetas, que aparecen en la parte superior de la ventana. El cuadro de texto que aparece en la parte inferior izquierda es el campo label, donde se ingresa el nombre del bloque. A pesar de que estos labels pueden ser modificados por el usuario, también pueden ser accedidos desde código ModL.
Código ModL	El programa ModL que hace que el bloque funcione. El programa lee información de los conectores, del cuadro de diálogo, y del ambiente del modelo para producir la salida que puede ser utilizada por otros bloques. Un bloque también puede consultar y controlar otros bloques del modelos a través de código ModL.
Icono	El icono que se ve en el modelo. Se puede dibujar usando el ambiente de dibujo de Extend o un programa de dibujo externo. Los iconos contienen la animación.
Animación	Parte del icono. Lo que se ve cuando se ejecuta la simulación con la opción de mostrar la animación activa. La animación puede mostrarse dentro del icono del bloque o también fuera del mismo.
Conectores	Los conectores de entrada y salida del bloque. Estos aparecen en el icono y transmiten información desde y hacia el código ModL. Los bloques también pueden transmitir información sin utilizar conectores, mediante mensajes de bloques y variables globales.
Texto de ayuda	El texto que aparece al hacer click en el icono de ayuda del cuadro de diálogo. La ayuda del bloque también está disponible en el menú <i>Help</i> .

Tabla 4.2: Partes de un bloque

La forma de acceder a la estructura de un bloque es a través de la opción *Open Block Structure* del menú *Define*, habiendo seleccionado previamente un bloque del modelo. La estructura del bloque consta de dos ventanas, la ventana de estructura y la ventana de cuadro de diálogo.

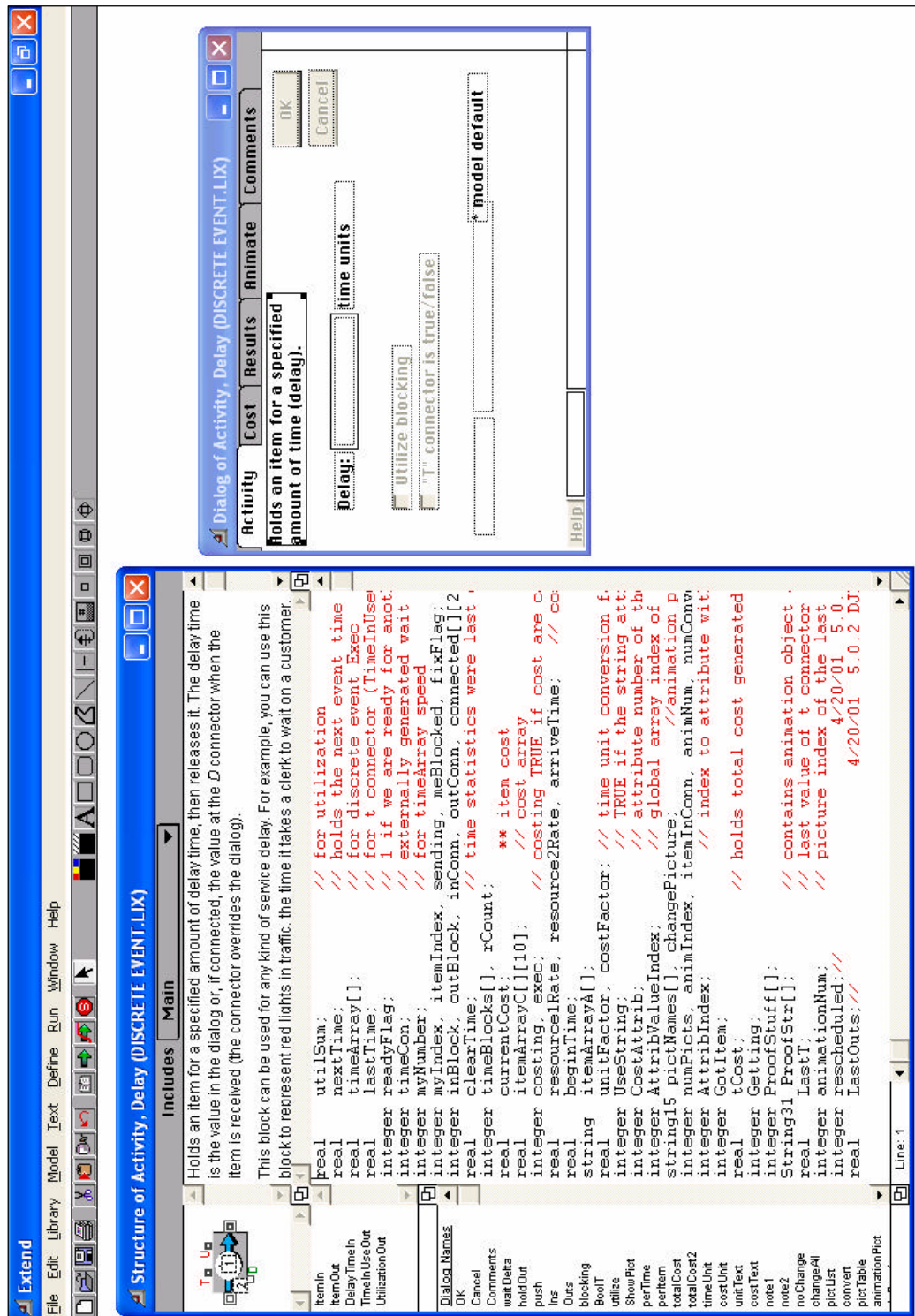


Figura 4.24: Estructura del bloque Activity, Delay.

La ventana izquierda es la de estructura, y la derecha la de cuadro de diálogo

La ventana de estructura está dividida en varios paneles. En la parte superior izquierda está el panel de icono, debajo de él el panel de conectores, debajo de éste el panel de variables. En la parte superior derecha está el panel de ayuda y debajo de él el panel de código. Arriba de todos estos paneles se encuentra el panel de inclusiones.

El panel de icono permite ingresar y editar el icono del bloque. Para crear el icono se pueden utilizar las herramientas de dibujo o pegar un dibujo existente. En este panel también se agregan los conectores del bloque y los objetos de animación.

El panel de conectores permite editar los nombres de los conectores para que representen de manera más acertada su función.

El panel de variables muestra las variables definidas para el cuadro de diálogo del bloque. También muestra las variables globales.

El panel de ayuda permite ingresar y editar el texto de ayuda del bloque. La ayuda del bloque es fácilmente accesible haciendo click en el botón de ayuda en la parte inferior izquierda del cuadro de diálogo de bloque.

El panel de código permite editar el código ModL del bloque. El editor provee ayuda para el desarrollo del código, como distintos colores según el tipo de sentencia, completado automático de los parámetros de funciones y resaltado de errores de compilación.

El panel de inclusiones permite abrir los archivos que se están incluyendo mediante sentencias #include.

La ventana de cuadro de diálogo contiene los botones, textos, y campos de entrada del cuadro de diálogo del bloque. Además, abajo a la izquierda, se encuentran el botón de ayuda y el campo label, donde se ingresa el nombre del bloque.

A continuación se describirán en más detalle los componentes de la estructura de un bloque.

Cuadro de diálogo

Cada bloque tiene un cuadro de diálogo asociado. Un cuadro de diálogo puede ser algo tan simple como un texto con botones de Aceptar y Cancelar, o algo tan complejo como sea necesario.

Un cuadro de diálogo contiene ítems de diálogo, un label del bloque, y un botón de ayuda. Cada ítem de diálogo tiene sus propias definiciones. Existen muchos tipos de ítems de diálogo, como se puede ver en la Figura 4.25.

Los cuadros de diálogo pueden tener lengüetas. Éstas son similares a páginas, y permiten agrupar los ítems de diálogo.

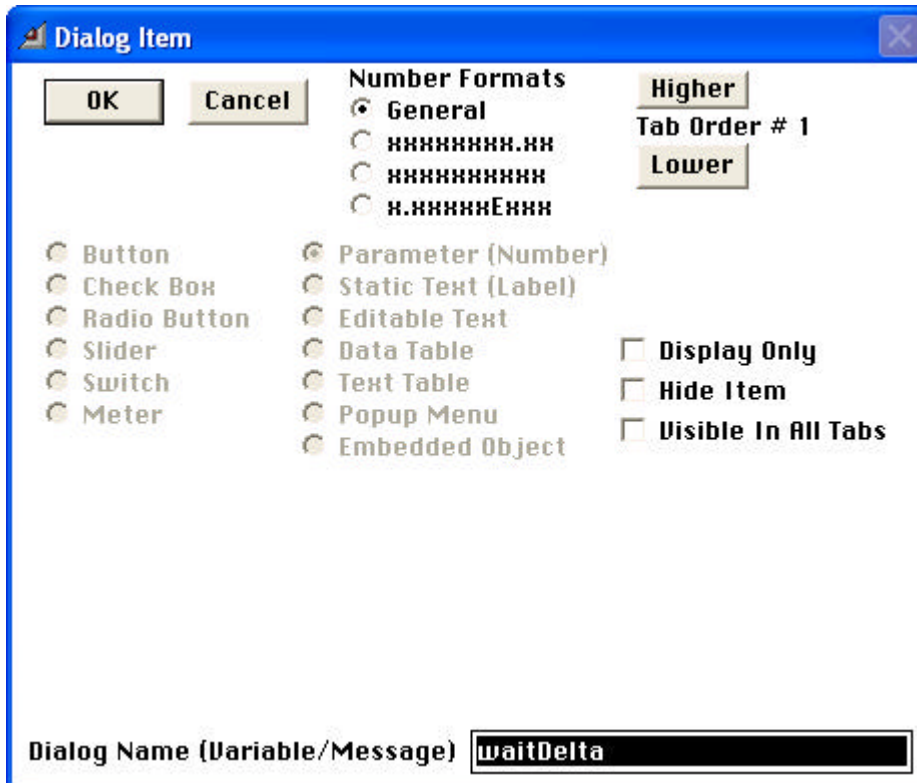


Figura 4.25: Configuración de un ítem de diálogo

Código ModL

El código ModL es similar a C. ModL es el lenguaje de programación interno de Extend. Cuando se construyen bloques en Extend usando ModL, el programa del bloque se compila en lenguaje de máquina. ModL es básicamente C con algunas extensiones y modificaciones.

La estructura del código ModL de los bloques incluidos en Extend es la siguiente. Las primeras líneas son declaraciones de los tipos de variables que se utilizan. Luego hay líneas con comentarios, que comienzan con «//». Luego de los comentarios, se encuentran las líneas de código que forman el programa, divididas en secciones. Cada sección es un manejador de mensajes o un procedimiento. Los manejadores de mensajes comienzan con «on xxx», donde xxx es el mensaje a manejar. Los procedimientos comienzan con «procedure yyy» o «type yyy», donde yyy es el nombre del procedimiento.

Los manejadores de mensajes le dicen a Extend que hacer en distintas circunstancias. Por ejemplo, el manejador «on Simulate» es ejecutado en cada paso de la simulación, el manejador «on InitSim» es ejecutado sólo una vez al inicio de la simulación.

Icono

El icono es la forma más común de representación de un bloque, ya que es lo que aparece en la ventana del modelo. El icono consiste de un dibujo y un conjunto de conectores.

Hay dos formas de crear un icono, dibujándolo con las herramientas de dibujo de Extend, o pegándolo de otra aplicación. Extend provee un ambiente de dibujo para quienes no disponen de un programa de dibujo. Si se posee un programa de dibujo, se puede crear allí el icono y luego pegarlo en Extend.



Figura 4.26: Panel de icono

1. Animación

Hay varias formas de darle animación a un bloque. Típicamente se anima el icono, pero también se puede animar fuera del icono. Los bloques pueden mostrar, ocultar o cambiar los patrones y colores de textos y formas; mover un dibujo o cambiar su tamaño; mostrar un nivel cambiante; mostrar una foto o película; o mover un dibujo a través de las líneas de una conexión. En la Figura 4.26 se pueden ver dos objetos de animación, son los rectángulos blancos numerados. Estos objetos pueden ser agregados utilizando la barra de herramientas. El código ModL interactúa con los objetos de animación para mostrar una animación durante la simulación.

2. Conectores

La mayoría de los bloques tienen conectores, que le permiten relacionarse con otros bloques. Los conectores se agregan al icono desde la barra de herramientas.

Cada conector tiene un nombre único en el bloque. Además, su nombre define si es un conector de entrada o de salida. Los nombres de los conectores de entrada terminan en In, los de salida en Out. Los nombres de los conectores se muestran en el panel de conectores.

Como se describió anteriormente, los conectores pueden ser de tipo ítem o valor. La barra de herramientas permite elegir el tipo de conector, e inclusive cambiar el tipo de un conector existente.

3. Texto de ayuda

El texto de ayuda puede ser modificado en el panel de ayuda. Se puede escribir lo que se desee, simplemente editando el texto. También se puede modificar el formato eligiendo el texto y usando las opciones del menú *Text*. El texto de ayuda aparece cuando se hace click en el botón de ayuda de la parte inferior izquierda del cuadro de diálogo de un bloque, y también está disponible desde el menú *Help*.

4.6.2. Introducción al código ModL

ModL es un lenguaje de programación muy similar al C. Para crear o modificar código ModL, es recomendable entender básicamente el lenguaje C.

4. Comparación entre ModL y C

Hay sólo una pequeña cantidad de diferencias entre ModL y C. Estas son las siguientes:

ModL	C
Insensible a mayúsculas y minúsculas	Sensible a mayúsculas y minúsculas
real o double	double
integer o long	long
string o str255 (cadena de hasta 255 caracteres)	typedef struct string {unsigned char length; unsigned char str[255];} string;
str15 (cadena de hasta 15 caracteres)	typedef struct string {unsigned char length; unsigned char str[15];} string;
str31 (cadena de hasta 31 caracteres)	typedef struct string {unsigned char length; unsigned char str[31];} string;
Comentarios: Sólo // hasta el final de la línea	Los comentarios pueden ser multilínea mediante /* */
Control de límites para vectores que produce mensajes de error si se excede el límite	No hay control de límites
Las funciones se declaran según el estándar ANSI	Las funciones pueden ser declaradas siguiendo el estándar ANSI o K&R
Las funciones y manejadores de mensajes pueden ser sobrecargados	No disponible en C, pero sí en C++
No se pueden usar expresiones dentro de otras expresiones. Por ejemplo, «a[i++] = 5;» debe ser escrito como «i++; a[i] = 5»	Se pueden usar expresiones dentro de otras expresiones
^ es usado como operador exponencial	^ es usado como operador XOR en expresiones lógicas
El operador + puede concatenar strings	Para concatenar strings: strcat(str1, «abc»);
Para convertir un número a string: stringVar = x; if (stringVar < «abc»)...	El equivalente en C es: ftoa(x, stringVar); if (strcmp(stringVar, «abc») < 0)...
Vectores dinámicos	Punteros
Listas encadenadas soportan estructuras de datos complejas	Estructuras
!= y <> son los operadores de desigualdad	!= es el operador de desigualdad
% y MOD son los operadores de módulo	% es el operador de módulo
El manejo de bits se realiza mediante funciones. Por ejemplo, BitAnd(n,m);	El manejo de bits se realiza mediante operadores. Por ejemplo, n&m;

Tabla 4.3: Diferencias entre ModL y C

5. Estructura general de un programa en ModL

Al igual que en C, los programas en ModL empiezan con las declaraciones de tipos y definiciones de constantes. Por ser declaradas al principio del código, son consideradas permanentes o estáticas. Son, por lo tanto, válidas en todo el código del bloque, a no ser que sean sobrecargas por una declaración de variable local.

Luego de estas declaraciones y definiciones, se encuentran las definiciones de funciones y procedimientos, y los manejadores de mensajes. Los procedimientos, funciones y manejadores de mensajes son sólo declaraciones, deben ser llamados para ser ejecutados. Los manejadores de mensajes interpretan mensajes provenientes de la simulación o del usuario. Extend ejecuta el manejador cuando un mensaje es enviado al bloque. Se declaran variables locales dentro de las funciones, los procedimientos y los manejadores de mensajes.

Las funciones y manejadores de mensajes pueden ser sobrecargados mediante redeclaraciones luego

de la primer declaración. Esto es útil ya que los archivos que se incluyen pueden tener funciones y manejadores de mensajes básicos que el programador necesite redefinir.

Como C, ModL ignora los espacios en blanco y la sangría. Es buena práctica utilizar sangría para hacer que el código sea más fácil de leer.

La forma de acceder a los datos de los conectores y los ítems de diálogo es a través de sus nombres. Los nombres de conectores e ítems de diálogo son usados como variables en ModL. Por ejemplo, para leer el valor de un conector de entrada llamado firstConIn:

```
int number;
number = firstConIn;
Para asignar un valor a un conector de salida llamado totalOut:
if (myNumber > 0)
    totalOut = 10.0;
```

Todos los conectores son variables de tipo real. Si se asigna un valor entero a un conector, Extend lo convertirá automáticamente a real. Esta conversión hace que la simulación sea más lenta. También se pueden pasar vectores de valores a través de los conectores.

La lectura y asignación de valores de los ítems de diálogo se hace en forma similar a los conectores, utilizando los nombres de los ítems. Por ejemplo, si se tiene un ítem de diálogo llamado numberOfRecords, se puede acceder a la siguiente forma:

```
int number;
number = numberOfRecords/100;
```

Distintos tipos de ítems de diálogo retornan distintos tipos de valores. Por ejemplo, los cuadros de texto retornan el valor ingresado, los botones de radio y checkboxes retornan verdadero o falso, y una tabla retorna un vector bidimensional de números reales o texto. Independientemente del tipo de valor que retornen, la forma de acceder a estos valores no varía.

Los manejadores de mensajes permiten a los bloques reaccionar antes distintos sucesos. Extend provee una gran cantidad de mensajes, pero sólo algunos de ellos son utilizados en la mayoría de los bloques. Hay dos tipos de mensajes, los de sistemas y los de cuadros de diálogo. Los mensajes de sistema son enviados a los bloques por el motor de Extend, los de cuadros de diálogo son enviados a los bloques cuando el usuario interactúa con el cuadro de diálogo de un bloque.

4.7. Soporte

Imagine That Inc. provee soporte para Extend de varias maneras. A continuación se detallan las mismas.

El sitio Web de Imagine That Inc. provee información detallada sobre formas de contactarse con la empresa. Se puede contactar a Imagine That Inc. en forma telefónica, por fax, por correo o por correo electrónico. También se pueden contactar a los revendedores locales de cada país o región. Estos revendedores realizan marketing, ventas, soporte técnico, actualizaciones y servicios de consultoría para productos Extend. En Argentina el revendedor local es la consultora S&T S.A.

A través del sitio Web de Imagine That Inc. también se puede acceder a un foro llamado Extend E-Xchange. Extend E-Xchange es un foro para usuarios de Extend, donde se puede hablar con otros usuarios de Extend, intercambiar sugerencias, realizar consultas, informarse de eventos futuros y compartir modelos y bloques.

Imagine That Inc. provee actualizaciones para las versiones de Extend periódicamente a través de su sitio Web. Estas actualizaciones contienen mejoras y correcciones de errores sobre versiones anteriores. La versión actual es la 6.0.4 que fue lanzada el 25 de Noviembre de 2003. Estas actualizaciones son gratuitas, si es que se posee la versión 6 de Extend.

Todas las versiones de Extend incluyen un manual de usuario en inglés. Este manual describe detalladamente todas las funcionalidades de todas las versiones, y su forma de utilización. Contiene tutoriales con ejemplos para aprender a utilizar el producto en forma práctica. Además del manual de usuario, Extend incluye una guía para el programador. Esta guía describe como programar bloques propios, utilizando el lenguaje de programación provisto por Extend, ModL. El revendedor en Argentina, la consultora S&T S.A., provee los manuales en español para la versión 6 de Extend.

Las versiones de Extend también incluyen una gran cantidad de modelos de ejemplo, que pueden ser analizados para aprender como se utilizan determinadas características de Extend. Estos modelos incluyen documentación en la ayuda del modelo, que facilitan la comprensión del mismo.

Existen un conjunto de empresas que, mediante alianzas con Imagine That Inc., proveen entrenamiento para Extend. Este entrenamiento puede ser tanto en un centro de educación de la empresa que provee el entrenamiento, como en el cliente que contrata el entrenamiento. Por ejemplo, James Dailey & Associates ofrece un programa donde colabora en el desarrollo del modelo del cliente, proveyendo su experiencia en el

uso de Extend, y comunicando el know-how que posee del desarrollo de modelos en Extend. Simulation Dynamics Inc. ofrece cursos clásicos, para diferentes niveles y diferentes tipos de modelos. 1Point2 es otra empresa que provee entrenamiento en Extend, mediante cursos, seminarios y servicios de consultoría. Las dos primeras empresas se encuentran en los Estados Unidos, la última en Francia.

Imagine That Inc. provee una versión de demostración de Extend en su sitio Web. La restricción de esta versión, que no tiene fecha de expiración, es que no permite guardar o imprimir modelos. Sin embargo, permite:

- Abrir y ejecutar modelos de cualquier tamaño.
- Cambiar los parámetros dentro de los bloques.
- Agregar nuevos bloques o modificar conexiones en modelos de menos de 25 bloques.
- Construir, pero no guardar, modelos de hasta 25 bloques.
- Crear nuevos bloques y guardarlos en librerías.

5. Caso de Estudio

El caso de estudio propuesto consiste en la simulación de una cola de un banco. Este banco abre los días de semana de 10 a 15 y tiene cinco cajeros. Para proveer un mejor servicio, el directorio del banco está evaluando la posibilidad de abrir el banco los sábados a la mañana, entre las 9AM y las 12PM.

Se cree que el tiempo entre arribos de clientes durante el sábado a la mañana será entre medio minuto y dos minutos, según la siguiente distribución de probabilidad:

Tiempo entre arribos (minutos)	Probabilidad
.5	.65
1	.15
1.5	.15
2	.05

Tabla 5.1: Distribución de probabilidad de tiempo entre arribos de clientes

El directorio está considerando que durante los sábados trabajen un cajero senior y dos junior. El cajero senior atiende más rápidamente a los clientes que los cajeros junior, debido a su experiencia. El tiempo de servicio para el cajero senior será entre medio minuto y 3 minutos y medio, según la siguiente distribución de probabilidad:

Tiempo de servicio de un cajero senior (minutos)	Probabilidad
.5	.05
1	.10
1.5	.20
2	.30
2.5	.20
3	.10
3.5	.05

Tabla 5.2: Distribución de probabilidad de tiempo de servicio de un cajero senior

El tiempo que le toma a los otros dos cajeros atender un cliente está estimado entre un minuto y cuatro minutos y medio, según la siguiente distribución de probabilidad:

Tiempo de servicio de un cajero junior (minutos)	Probabilidad
1	.05
1.5	.15
2	.20
2.5	.30
3	.10
3.5	.10
4.5	.05

Tabla 5.3: Distribución de probabilidad de tiempo de servicio de un cajero junior

El banco utiliza una regla de prioridad FCFS¹ para atender a sus clientes.

El directorio desea utilizar la simulación para determinar el tiempo promedio de espera de los clientes durante el sábado a la mañana.

Se analizará la resolución mediante tres métodos. Primero se resolverá el problema en forma manual, luego utilizando Excel, y por último utilizando Extend. En este último caso también se verán algunas de las ventajas y posibilidades que provee el uso de Extend, en comparación con la opción manual o el uso de Excel.

5.1. Solución en Forma Manual

Antes de desarrollar el modelo, debemos determinar si se satisface la condición de equilibrio, que consiste en que el número de clientes que pueden ser atendidos por hora sea mayor al número de clientes que arriban al sistema por hora.

Para calcular el número de clientes que arriban por hora:

$$E(\text{Tiempo entre arribos}) = .5(.65) + 1(.15) + 1.5(.15) + 2(.05) = .80 \text{ minutos}$$

Esto equivale a $1/.80 = 1.25$ clientes por minuto o $60 \times 1.25 = 75$ clientes por hora.

Para calcular el número de clientes atendidos por hora:

$$E(\text{Tiempo de servicio senior}) = .5(.05) + 1(.10) + 1.5(.20) + 2(.30) + 2.5(.2) + 3(.10) + 3.5(.05) = 2 \text{ minutos}$$

$$E(\text{Tiempo de servicio junior}) = 1(.05) + 1.5(.15) + 2(.20) + 2.5(.3) + 3(.10) + 3.5(.10) + 4(.05) + 4.5(.05) = 2.5 \text{ minutos}$$

Por lo tanto, el cajero senior podrá servir $60/2 = 30$ clientes por hora, y cada uno de los junior podrá servir $60/2.5 = 24$ clientes por hora. El promedio total de clientes que pueden ser atendidos por hora es de $30 + 24 \times 2 = 78$ clientes por hora. Dado que este número es mayor al número de clientes que arriban por hora, se alcanzará la condición de equilibrio.

El primer paso en el desarrollo de la simulación es determinar el mapeo de números aleatorios. Las siguientes tablas muestran este mapeo.

Tiempo entre arribos (minutos)	Números aleatorios
.5	00-64
1	65-79
1.5	80-94
2	95-99

Tabla 5.4: Mapeo de tiempo entre arribos de clientes

Tiempo de servicio de un cajero senior (minutos)	Números aleatorios
.5	00-04
1	05-14
1.5	15-34
2	35-64
2.5	65-84
3	85-94
3.5	95-99

Tabla 5.5: Mapeo de tiempo de servicio de un cajero senior

Tiempo de servicio de un cajero junior (minutos)	Números aleatorios
1	00-04
1.5	05-19
2	20-39
2.5	40-69
3	70-79
3.5	80-89
4	90-94
4.5	95-99

Tabla 5.6: Mapeo de tiempo de servicio de un cajero junior

La siguiente tabla muestra los resultados de una simulación de veinte clientes usando números aleato-

rios de la columna cuatro de la tabla del Anexo A (se utilizan los primeros dos dígitos de cada número) para determinar el tiempo entre arribos de clientes y números aleatorios de la columna cinco para determinar el tiempo de servicio.

Cliente	Número Aleatorio	Tiempo de Arribo	Número Aleatorio	Cajero Senior		Cajero Junior 1		Cajero Junior 2		Tiempo de Espera
				Inicio	Fin	Inicio	Fin	Inicio	Fin	
1	89	1.5	63	1.5	3.5					0
2	88	3	46			3	5.5			0
3	90	4.5	86	4.5	7.5					0
4	26	5	00					5	6	0
5	79	6	56			6	8.5			0
6	55	6.5	67					6.5	9	0
7	26	7	59	7.5	9.5					0.5
8	16	7.5	28			8.5	10.5			1
9	40	8	79					9	12	1
10	65	9	64	9.5	11.5					0.5
11	61	9.5	33			10.5	12.5			1
12	68	10.5	81	11.5	14					1
13	75	11.5	17					12	13.5	0.5
14	65	12.5	63			12.5	15			0
15	08	13	66					13.5	16	0.5
16	31	13.5	27	14	15.5					0.5
17	20	14	04			15	16			1
18	40	14.5	34	15.5	17					1
19	99	16.5	07			16.5	18			0
20	09	17	69	17	19.5					0

Tabla 5.7: Ejecución de la simulación

La simulación procede de la forma siguiente:

- Un cliente irá al cajero senior si está libre.
- Si el Cajero Senior no está libre, el cliente irá al Cajero Junior 1.
- Si tanto el Cajero Senior como el Cajero Junior 1 están ocupados, el cliente irá al Cajero Junior 2.
- Si todos los cajeros están ocupados, el cliente esperará en la cola hasta que un cajero se desocupe.
- El tiempo de espera consiste en el tiempo que un cliente pasa en la cola antes de que comience el servicio. Es la diferencia entre el tiempo en el que comienza el servicio y el tiempo de arribo.

Se examinarán algunos de los arribos para entender como se lleva a cabo la simulación. Para el cliente 1, el número aleatorio usada para determinar el tiempo entre arribos es 89. Usando el mapeo de la tabla 5.4, esto corresponde a un tiempo entre arribos de 1.5 minutos. Por lo tanto, el cliente 1 arriba en el minuto 1.5. Como todos los cajeros están libres, lo atiende el Cajero Senior. El número aleatorio para el tiempo de servicio del cliente 1 es 63. El mapeo de la tabla 5.5 corresponde a un tiempo de servicio de 2 minutos. Por lo tanto el servicio comienza en el minuto 1.5 (cuando el cliente llega), y termina en el minuto 3.5 (tiempo de arribo de 1.5 más tiempo de servicio de 2). Por lo tanto, el Cajero Senior está ocupado atendiendo al cliente 1 hasta el minuto 3.5.

El tiempo entre arribos para el cliente 2 se determina con el número aleatorio 88, que según el mapeo corresponde a 1.5 minutos. Dado que el cliente 1 arribó en el minuto 1.5, el cliente 2 arriba en el minuto 3. Ya que el Cajero Senior está ocupado en el minuto 3, el Cajero Junior 1 atiende al cliente 2. El número aleatorio para el tiempo de servicio es 46, que según el mapeo de la tabla 5.6 corresponde a 2.5 minutos. Por lo tanto, el cliente 2 completará su servicio en el minuto 5.5 (tiempo de arribo de 3 más tiempo de servicio de 2.5), y el Cajero Junior 1 estará ocupado hasta ese momento.

Este proceso se repite para todos los clientes. Dado que los dos primeros clientes fueron atendidos inmediatamente, sus tiempos de espera fueron de 0 minutos. Para ilustrar el caso de un cliente que debe esperar, se analiza el cliente 7, que arriba en el minuto 7. En ese momento todos los servidores están ocupados. El primer cajero en desocuparse es el Cajero Senior, en el minuto 7.5. Por lo tanto, el servicio

comienza en el minuto 7.5 y el tiempo de espera para el cliente 7 es de 0.5 minutos (diferencia entre el momento en que comienza el servicio 7.5 y el momento de arribo 7).

La forma intuitiva para calcular el promedio de tiempos de espera sería simplemente sumar los valores de los tiempos de espera y dividir el resultado por el número de clientes. Esto daría como resultado un promedio de $8.5/20 = .425$ minutos. Pero dado que la mayoría de las simulaciones de colas comienzan vacías, hay un cierto tiempo de inicialización asociado a los primeros clientes hasta que el sistema alcanza un estado de equilibrio. Por lo tanto, los primeros clientes deberían ser excluidos del cálculo del promedio.

Si los primeros 10 clientes son excluidos, entonces el tiempo de espera de los últimos 10 clientes podría ser usado para estimar una espera promedio de $5.5/10 = .55$ minutos. Por supuesto, el verdadero promedio puede ser muy diferente de .425 o .55 minutos. Un conjunto de números aleatorios distinto probablemente daría como resultado un valor diferente de tiempo de espera promedio. Realizar una simulación de 20 clientes no permite estimar con gran precisión el valor de tiempo de espera promedio.

Si se quiere realizar una simulación para 10000, 100000 o inclusive 100 clientes, el método de solución en forma manual no resulta útil. Para ello se pueden utilizar los métodos que se describen a continuación, Excel y Extend.

5.2. Solución Utilizando Excel

Se pueden construir hojas de cálculos en Excel para realizar simulaciones que en forma manual serían tediosas o imposibles. A continuación se resolverá el caso del banco utilizando Excel para simular el arribo de 10000 clientes. Dos puntos importantes son que tanto el tiempo que el cliente espera en la cola y el tiempo que pasa en el sistema se calculan en la misma hoja de cálculos. El hecho de realizar la simulación con 10000 clientes hace que el tiempo de inicialización hasta que el sistema alcanza el estado de equilibrio tenga un efecto imperceptible y pueda ser ignorado.

Client e	Número Aleatorio	Tiempo de Arribo	Número Aleatorio	Cajero Senior	Cajero Junior 1	Cajero Junior 2	Tiempo de Espera	Tiempo de Espera en	Tiempo entre arribos
				Inicio	Fin	Inicio	Fin	Inicio	Fin
1	0.94	1.6	0.59	1.6	3.5				
2	0.35	2.0	0.21			2	4		
3	0.11	2.6	0.32			2.5	4.5		
4	0.47	3.0	0.31	3.5	5				
5	0.30	3.5	0.15			4	5.5		
6	0.78	4.5	0.62			4.5	7		
7	0.08	5.0	0.94	5	8				
8	0.21	5.5	0.66			5.5	8		
9	0.95	7.0	0.31					7	9
10	0.16	7.5	0.49	8	10				
11	0.67	9.0	0.89			9	12.5		
12	0.49	9.5	0.26					9.5	11.5
13	0.30	10.0	0.46	10	12				
14	0.93	11.5	0.94					11.5	15.0
15	0.61	13.0	0.32	13	14.5				
16	0.35	13.5	0.30			13.5	15.5		
17	0.25	14.0	0.04	14.5	15				
18	0.63	14.5	0.86	15	18				
19	0.75	15.5	0.31			15.5	17.5		
20	0.34	16.0	0.98					16	20.5
21	0.35	16.5	0.47			17.5	20		
22	0.14	17.0	0.22	18	19.5				
23	0.79	18.0	0.56	19.5	21.5				
24	0.99	20.0	0.51			20	22.5		

Figura 5.1: Simulación del banco utilizando Excel

Las fórmulas que componen esta hoja de cálculo son las siguientes:

- Celdas A8 a A10007: Número de cliente. Se calcula sumando 1 al número de la celda superior. $A8=A7+1$
- Celdas B8 a B10007: Número aleatorio. Se utiliza la función RAND() para generarlos. $B8=RAND()$
- Celdas C8 a C10007: Tiempo de arribo. Se calcula mediante la búsqueda del número aleatorio de la celda de la izquierda en la tabla de Tiempo entre arribos de las celdas M4:N7. Este valor corresponde al tiempo entre arribos, por lo tanto se le suma el valor de la celda superior.
 $C8 = C7+VLOOKUP((B8),\$M\$4:\$N\$7,2)$
- Celdas D8 a D10007: Número aleatorio. Se utiliza la función RAND() para generarlos. $D8=RAND()$
- Celdas E8 a E10007: Tiempo de inicio de servicio del Cajero Senior. Chequea si en el tiempo de arribo el Cajero Senior está libre. Si está libre, el tiempo de inicio será el de arribo. Si el Cajero Senior está ocupado, la fórmula chequea si los otros dos cajeros están ocupados, y se liberarán después del Cajero Senior. Si ese es el caso, el Cajero Senior atenderá a este cliente en cuanto se desocupe, y ese será el tiempo de inicio de servicio. Si no, otro de los cajeros atenderá a este cliente y esta celda quedará en blanco. $E8=IF(MAX(\$F\$7:F7)\leq C8,C8,IF(AND(MAX(\$F\$7:F7)\leq MAX(\$H\$7:H7),MAX(\$F\$7:F7)\leq MAX(\$J\$7:J7)),MAX(\$F\$7:F7),»»))$
- Celdas F8 a F10007: Tiempo de fin de servicio del Cajero Senior. Chequea si el Cajero Senior es el que está atendiendo al cliente. Si es así, suma el valor del tiempo de inicio de servicio al de la búsqueda del número aleatorio de la columna D en la tabla de Tiempo de Servicio del Cajero Senior. $F8=IF(E8=»»»,»»,E8+VLOOKUP(D8,\$M\$11:\$N\$17,2))$
- Celdas G8 a G10007: Tiempo de inicio de servicio del Cajero Junior 1. Chequea si el Cajero Senior está atendiendo al cliente. Si es así, esta celda queda en blanco. Caso contrario realiza una operación similar a la de la celda de Tiempo de inicio de servicio del Cajero Senior. $G8=IF(E8<>»»»,»»,IF(MAX(\$H\$7:H7)\leq C8,C8,IF(AND(MAX(\$H\$7:H7)\leq MAX(\$F\$7:F7),MAX(\$H\$7:H7)\leq MAX(\$J\$7:J7)),MAX(\$H\$7:H7),»»)))$
- Celdas H8 a H10007: Tiempo de fin de servicio del Cajero Junior 1. Opera en forma similar a la celda de Tiempo de fin de servicio del Cajero Senior, buscando sus valores en la tabla de Tiempo de Servicio de un Cajero Junior. $H8=IF(G8=»»»,»»,G8+VLOOKUP(D8,\$M\$20:\$N\$27,2))$
- Celdas I8 a I10007: Tiempo de inicio de servicio del Cajero Junior 2. Opera en forma similar a la celda de Tiempo de inicio de servicio del Cajero Junior 1, chequeando que tanto el Cajero Senior como el Cajero Junior 1 no estén atendiendo a este cliente. $I8=IF(OR(E8<>»»»,G8<>»»»),»»,MAX(C8,MAX(\$J7:J\$8)))$
- Celdas J8 a J10007: Tiempo de fin de servicio del Cajero Junior 2. Opera en forma similar a la celda de Tiempo de fin de servicio del Cajero Junior 1. $J8=IF(I8=»»»,»»,I8+VLOOKUP(D8,\$M\$20:\$N\$27,2))$
- Celdas K8 a K10007: Tiempo de espera en cola. Calcula la diferencia entre el tiempo de inicio del cajero que atiende al cliente y el tiempo de arribo del cliente. $K8=MAX(E8,G8,I8)-C8$
- Celdas L8 a L10007: Tiempo de espera en el sistema. Calcula la diferencia entre el tiempo de fin del cajero que atiende al cliente, es decir, el momento en que el cliente se va del sistema, y el tiempo de arribo del cliente. $L8=MAX(F8,H8,J8)-C8$
- Columnas M y N: Tablas de mapeo de números aleatorios. Se utiliza para relacionar los números aleatorios con los tiempos entre arribos y tiempos de servicio.
- Celda F1: Tiempo de espera promedio en cola. $F1=AVERAGE(K8:K10007)$
- Celda F2: Tiempo de espera promedio en sistema. $F2=AVERAGE(L8:L10007)$

A partir de los datos existentes, se puede calcular el número promedio de clientes en la cola y en el sistema. Estos promedios se obtienen multiplicando la frecuencia de arribos por el resultado correspondiente de la simulación, tiempo de espera promedio en la cola y en el sistema.

Dado que para este problema el tiempo entre arribos es .80 minutos, la frecuencia de arribos es $1/.80 = 1.25$ clientes por minuto. Por lo tanto, se obtienen los siguientes valores:

- Tiempo de espera promedio en cola: 3.686 minutos.
- Tiempo de espera promedio en el sistema: 5.998 minutos.
- Número promedio de clientes esperando en cola: $1.25 * 3.686 = 4.6075$ clientes.
- Número promedio de clientes en el sistema: $1.25 * 5.998 = 7.4975$ clientes.

Excel permite ejecutar la simulación cuantas veces sea necesario, mediante la tecla F9, que modifica los valores de las variables aleatorias. Excel también permite modificar el número de clientes simplemente arrastrando algunas fórmulas y modificando las que calculan los promedios.

Sin embargo, si se desea agregar un nuevo cajero, se tendrían que modificar todas las fórmulas que afectan a los cajeros, lo cual es una operación muy compleja, y que genera errores con facilidad. Extend soluciona esto ya que no es necesario escribir fórmulas complejas, lo cual hace que tanto el desarrollo del modelo como su mantenimiento sea mucho más fácil y rápido.

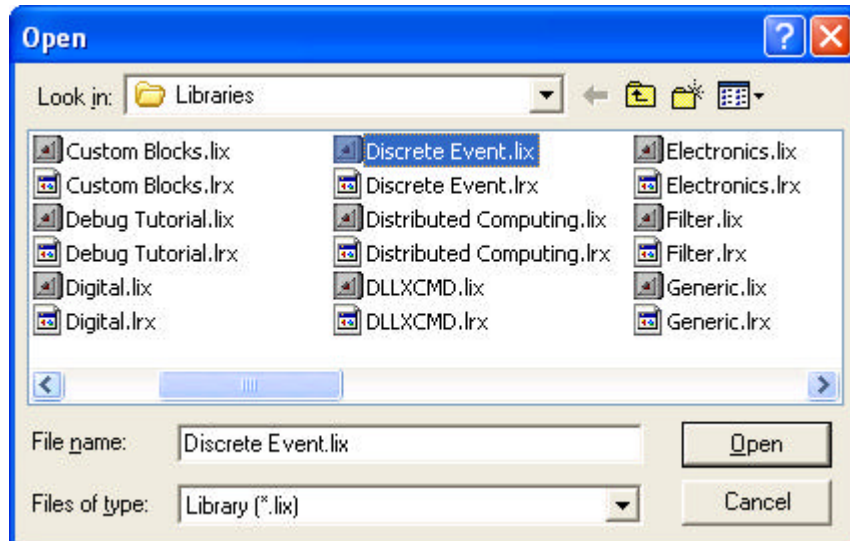


Figura 5.2: Abrir librerías

Una vez abiertas las librerías, se deben desplegar sus ventanas en la pantalla. Esto se hace mediante la opción *Open Library Window* de cada librería en el menú *Library*.

El primer bloque que debe agregarse al modelo es el *Executive*, ya que la simulación es de eventos discretos. Este bloque controla el tiempo en este tipo de simulaciones.



Este bloque debe ir a la izquierda de todos los demás del modelo. Se puede agregar arrastrándolo desde la ventana de la librería o desde el menú *Library, Discrete Event, Executive*.

El segundo bloque que se debe agregar es el que genera los clientes. Este bloque se denomina *Generator* y también se encuentra en la librería *Discrete Event*.



El bloque *Generator* tiene un conector a la derecha y cuatro abajo. El de la derecha es la salida del bloque, los de abajo son las entradas correspondientes. Haciendo doble click en el bloque se abre el cuadro de diálogo con las opciones del bloque.

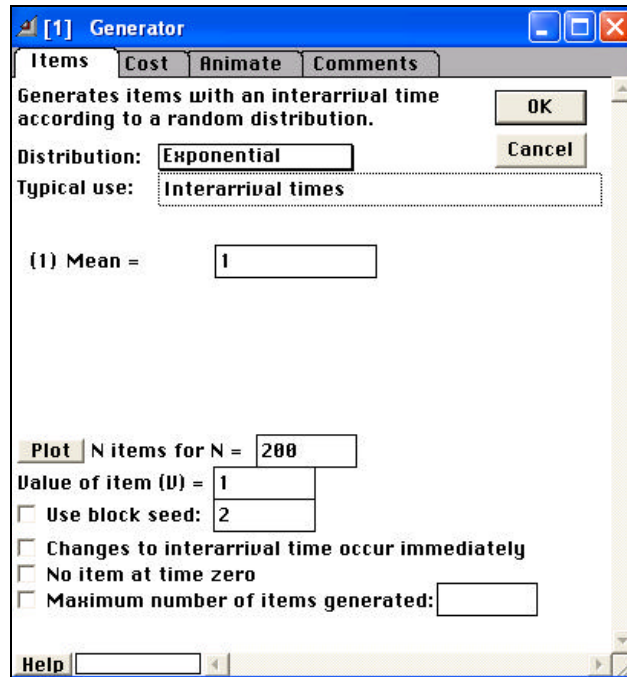


Figura 5.4: Cuadro de diálogo del bloque *Generator*

El bloque *Generator*, en forma predeterminada, tiene una distribución exponencial. Dado que en el caso del banco los clientes arriban con una distribución discreta, se debe modificar de exponencial a tabla empírica, donde se ingresan los datos. El cuadro debe como en la Figura 5.5.

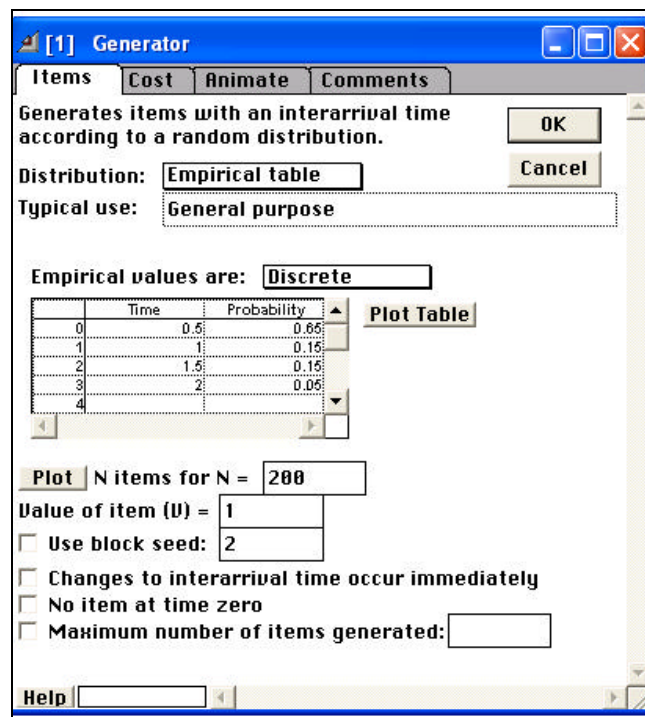


Figura 5.5: Cuadro de diálogo del bloque *Generator* para el problema del banco

Hay varias lengüetas en el cuadro de diálogo. En la de animación se puede elegir el icono que representará a los clientes generados. Dado que el modelo es un banco, se debe seleccionar *Person*, para que los iconos sean personas.

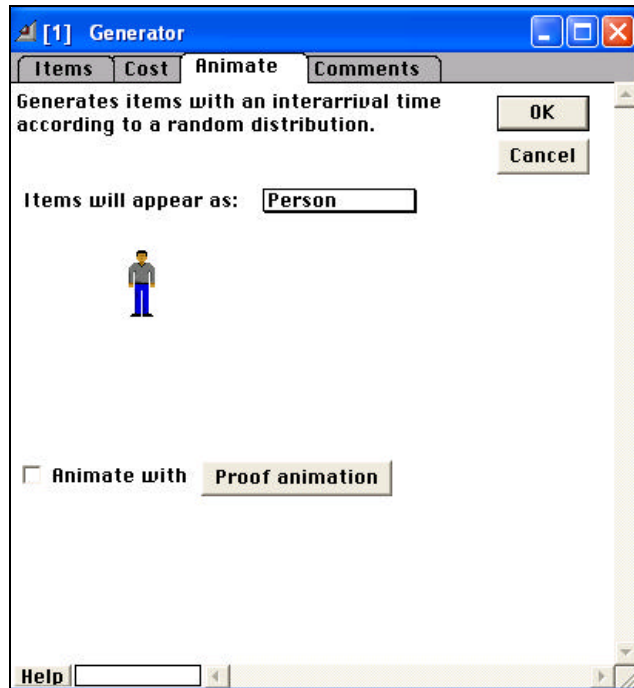


Figura 5.6: Cuadro de diálogo de la lengüeta de animación del bloque *Generator* para el problema del banco

El siguiente bloque a agregar es el correspondiente a la cola. Dado que la cola tiene una prioridad FIFO, se debe elegir el bloque *Queue, FIFO* de la librería *Discrete Event*. Este bloque debe ser agregado a la derecha del *Generator*. El resultado se puede ver en la Figura 5.7.

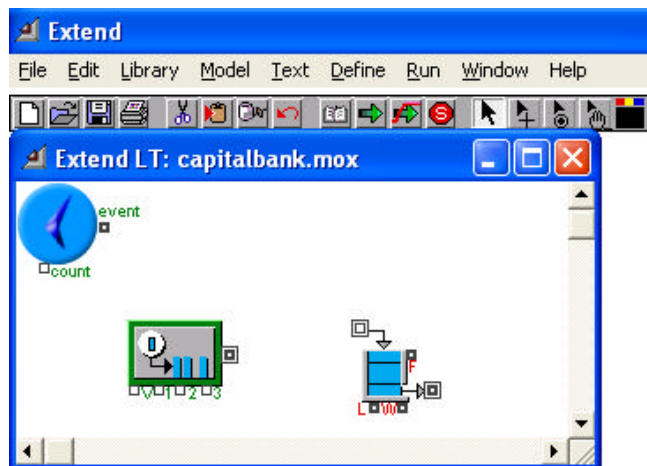


Figura 5.7: Modelo del banco con un bloque *Executive*, uno *Generator* y uno *Queue, FIFO*

El bloque *Queue, FIFO* tiene un conector de entrada en la esquina superior izquierda, y uno de salida a la derecha. También tiene tres conectores de salida, F, L y W que proveen información que puede ser útil para el analista. El conector F genera como salida 1 si la cola está llena, 0 en caso contrario. El conector L informa el número de ítems en cola. El conector W informa el tiempo de espera para los ítems que dejan la cola. El cuadro de diálogo de la cola permite modificar algunos parámetros, como el tamaño máximo.

El siguiente paso es conectar el bloque *Generator* con el bloque *Queue, FIFO*. Esto se consigue haciendo click en el conector de la derecha del bloque *Generator*, y manteniendo el botón apretado. Ahora se debe arrastrar el mouse hasta llegar al conector de la izquierda del bloque *Queue, FIFO*. Una vez allí soltar el botón del mouse, y la conexión quedará hecha.

Ahora se deben agregar tres bloques, uno por cada cajero. Estos bloques serán del tipo *Activity, Delay* de la librería *Discrete Event*. El bloque *Activity, Delay* tiene un conector de entrada a la izquierda y uno de salida a la derecha. Además, tiene un conector de salida T que informa el tiempo que el bloque lleva en uso, y un conector de salida U que muestra la utilización del bloque. El bloque tiene un tiempo de servicio predeterminado de una unidad de tiempo, que puede ser modificado por un valor fijo. Si el tiempo de servicio es variable, se utiliza otro bloque mediante el conector D, que es de entrada.

Los tiempos de servicio para el modelo del banco deben seguir las distribuciones del Cajero Senior y los Cajeros Junior, por lo tanto se necesitan dos generadores de números aleatorios. Estos se encuentran en la librería *Generic*, bajo el nombre de *Input Random Number*. El siguiente paso es agregar dos de estos bloques al modelo.

Una vez hecho esto, se deben realizar las conexiones entre la cola y cada uno de los tres servidores, y luego entre los servidores y los generados de números aleatorios correspondientes. Esto genera el modelo de la Figura 5.8.

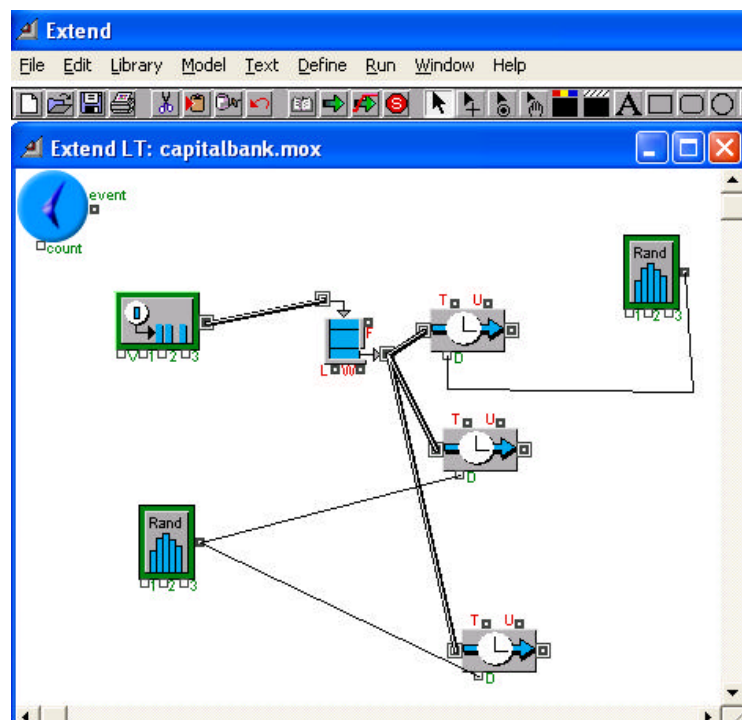


Figura 5.8: Modelo del banco con tres servidores y dos generadores de números aleatorios

El generador de números aleatorios de la parte superior derecha será utilizado para generar los tiempos de servicio del Cajero Senior, mientras que el de la parte inferior izquierda será utilizado para los dos Cajeros Junior.

Luego de hacer las conexiones, se deben definir las distribuciones de los tiempos de servicio en los bloques *Input Random Number*. En el cuadro de diálogo de estos bloques se pueden elegir un gran número de distribuciones, la utilizada para este caso será *Empirical Table*. Las Figuras 5.9 y 5.10 muestran el resultado de completar estos datos para el *Input Random Number* correspondiente al Cajero Senior y a los Cajeros Junior respectivamente.

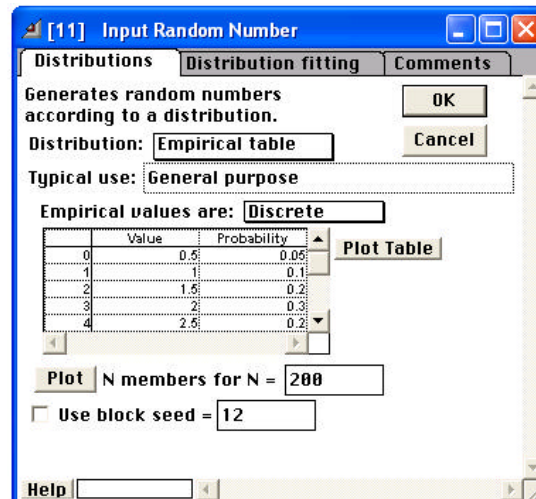


Figura 5.9: Cuadro de diálogo del bloque *Input Random Number* para el Cajero Senior

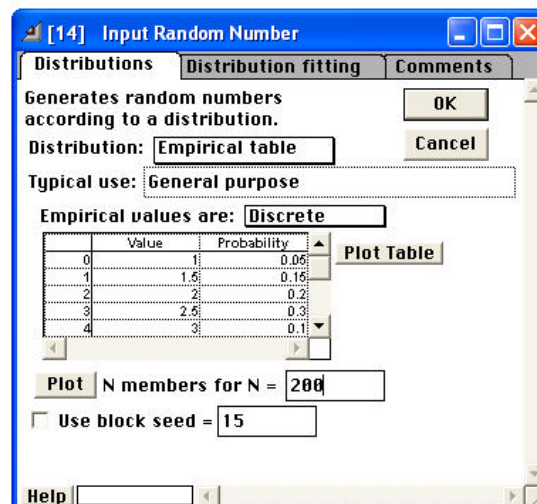


Figura 5.10: Cuadro de diálogo del bloque *Input Random Number* para los Cajeros Junior

El siguiente paso es agregar un bloque *Exit*, que se utiliza para indicar que la simulación está terminando. Los bloques *Exit* se encuentran en la librería *Discrete Event*. Hay dos tipos, *Exit* y *Exit(4)*, la diferencia es que *Exit(4)* permite hasta cuatro conexiones de entrada. Dado que se deben conectar tres servidores al bloque, se agrega un bloque *Exit(4)*, y se conectan los tres bloques *Activity*, *Delay* a él.

En este punto el modelo ya está completo. Sin embargo, se agregará un bloque *Plotter*, de la librería *Plotter*, para almacenar y mostrar el número de personas esperando en la cola. El tipo de *Plotter* que se utilizará es *Plotter, Discrete Event*. Luego se conecta la salida L de la cola al *Plotter*. Esto da como resultado el modelo de la Figura 5.11.

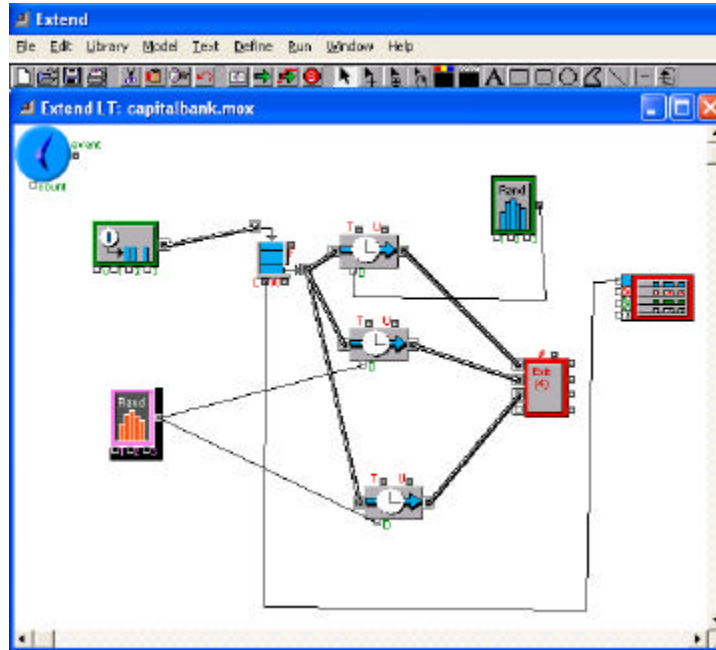


Figura 5.11: Modelo del banco final

Ahora se puede ejecutar la simulación. Para hacerlo, ingresar a *Run, Simulation Setup*. Allí se puede especificar la duración de la simulación, que aquí será de 30000 unidades de tiempo. Dado que en toda la simulación se utiliza la misma unidad de tiempo, que es minutos, no hace falta especificar aquí cuál es; se puede utilizar la unidad de tiempo genérica. Ahora ejecutar el comando *Run Now*, que inicia la simulación. La salida del Plotter se muestra durante la ejecución, y el resultado final es el de la Figura 5.12, donde se puede ver la evolución de la longitud de la cola en el tiempo.

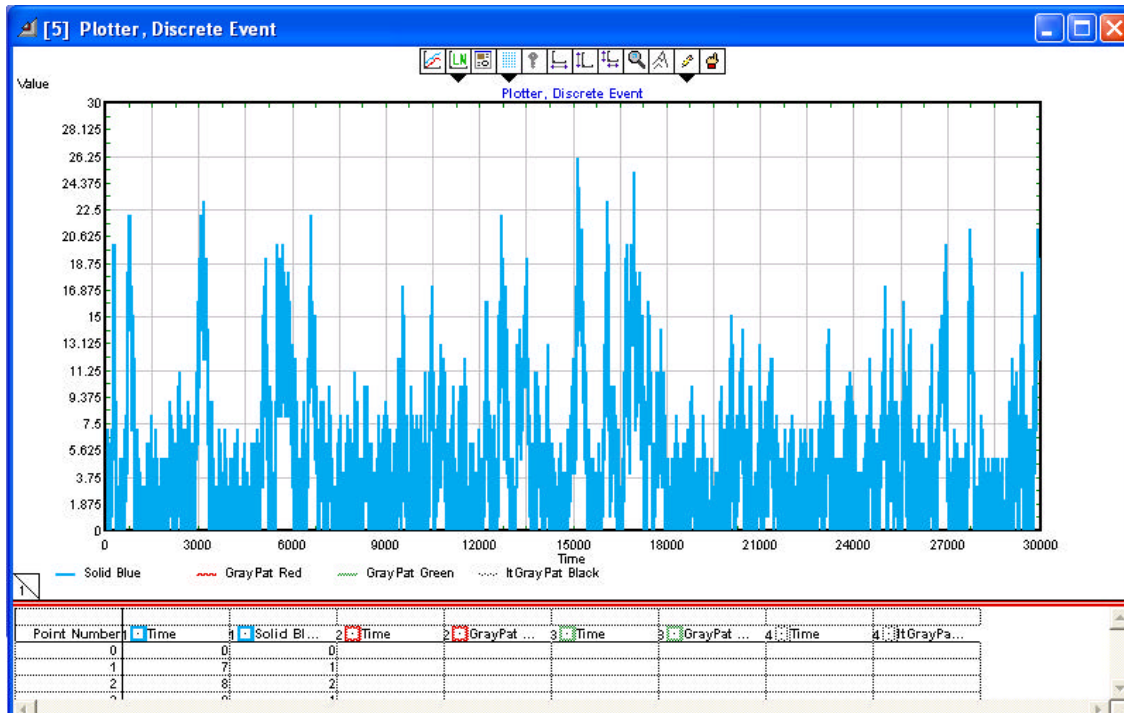


Figura 5.12: Resultado del Plotter del modelo

Para ver el resumen de los resultados de la ejecución, se pueden analizar los cuadros de diálogo de distintos bloques.

El cuadro de diálogo de la cola contiene las estadísticas de la cola:

- Longitud promedio de la cola: 4.4591 personas.
- Espera promedio en la cola: 3.5707 minutos.
- Longitud máxima de la cola: 27 personas.
- Espera máxima en la cola: 20 minutos.
- Costo: 0, ya que no fue especificado en el problema.
- Longitud de la cola (al finalizar la simulación): 14 personas.
- Arribos a la cola: 37456 personas.
- Partidas de la cola: 37442 personas.
- Utilización de la cola¹: 79.01%

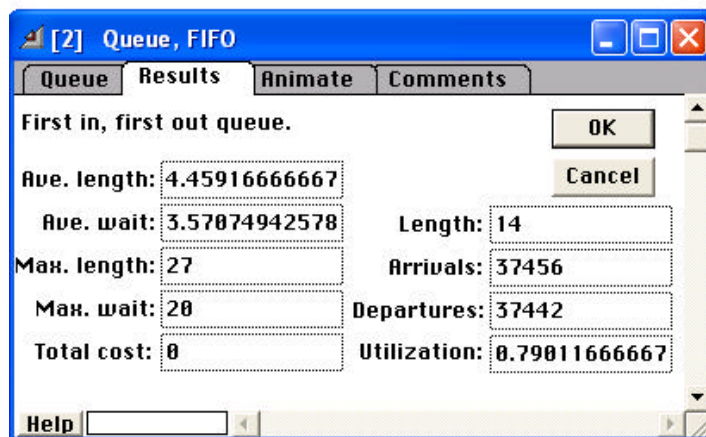


Figura 5.13: Cuadro de diálogo de resultados de la cola

El cuadro de diálogo de cada bloque *Activity, Delay* contiene información acerca de los cajeros. Para el Cajero Senior:

- Arribos al Cajero Senior: 14627 personas.
 - Partidas del Cajero Senior: 14626 personas.
- Utilización: 97.43%

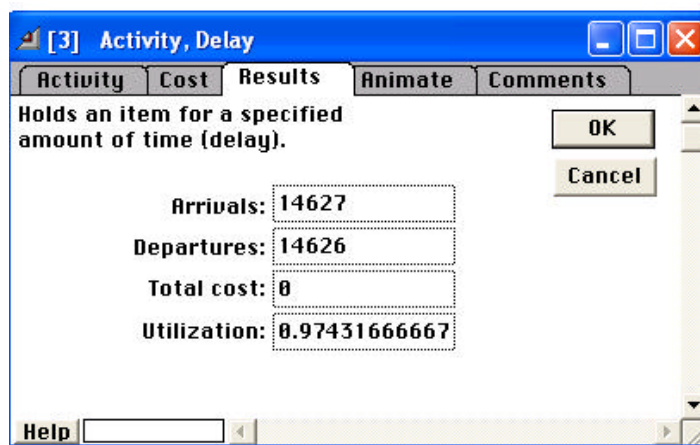


Figura 5.14: Cuadro de diálogo de resultados del bloque *Activity, Delay* para el Cajero Senior

Para el Cajero Junior 1 los resultados son 11561 arribos y 11560 partidas, con una utilización del 96.12%. Para el Cajero Junior 2 los resultados son 11254 arribos y 11253 partidas, con una utilización del 93.92%.

El cuadro de diálogo del bloque *Exit* contiene información acerca de los clientes atendidos por cada cajero, y el total de clientes atendidos:

- El Cajero Senior atendió 14626 clientes.
 - El Cajero Junior 1 atendió 11560 clientes.
 - El Cajero Junior 2 atendió 11253 clientes.
- Total de clientes atendidos: 37439 personas.

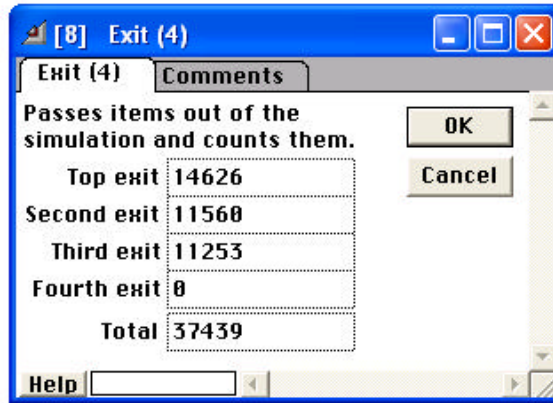


Figura 5.15: Cuadro de diálogo de resultados del bloque *Exit*

Como se puede ver, Extend provee una gran facilidad para el desarrollo de modelos. Se evitan tanto los problemas que implica la resolución manual de un modelo, como los que implica la utilización de Excel. A continuación se agregará un cajero más al modelo. Esta operación implicaría la modificación de la mayoría de las fórmulas en Excel, lo cual es una tarea que demanda mucho tiempo y puede generar errores.

El agregar un cajero al modelo es una tarea muy simple en Extend. Suponiendo que sea un Cajero Junior, lo único que se debe hacer es agregar un bloque *Activity*, *Delay* al modelo, y conectar a los bloques *Queue FIFO*, *Input Random Number* *Exit* que corresponden. El resultado se puede ver en la Figura 5.16.

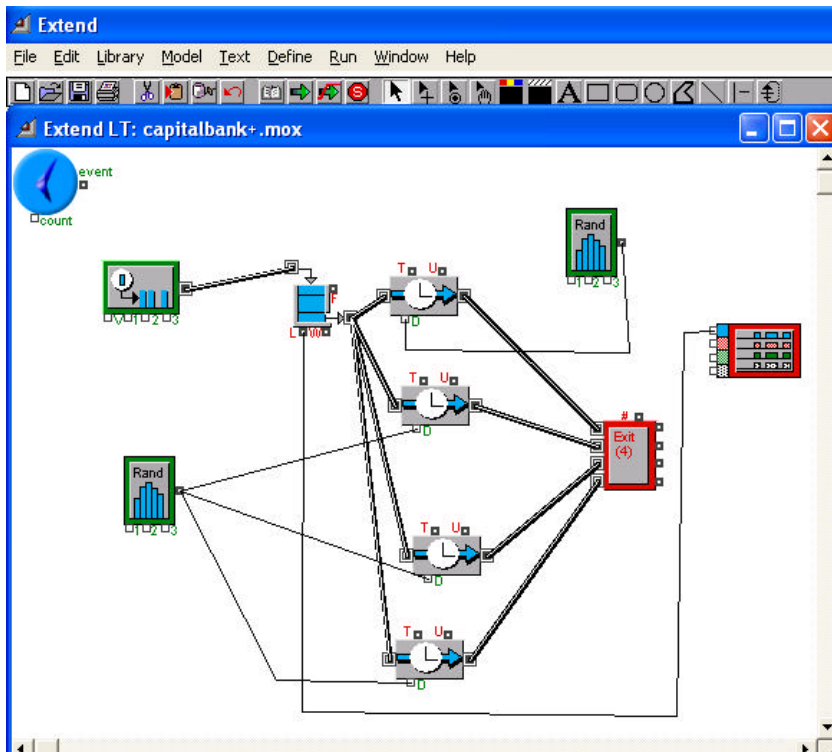


Figura 5.16: Modelo del banco con un Cajero Senior y tres Cajeros Junior

Por último, se pueden comparar los resultados del modelo antes y después de agregar un Cajero Junior. El resultado de dicha comparación se puede ver en la Figura 5.17. El cuadro de resultados de la izquierda corresponde al modelo antes de agregar un cajero, y el de la derecha después de hacerlo.

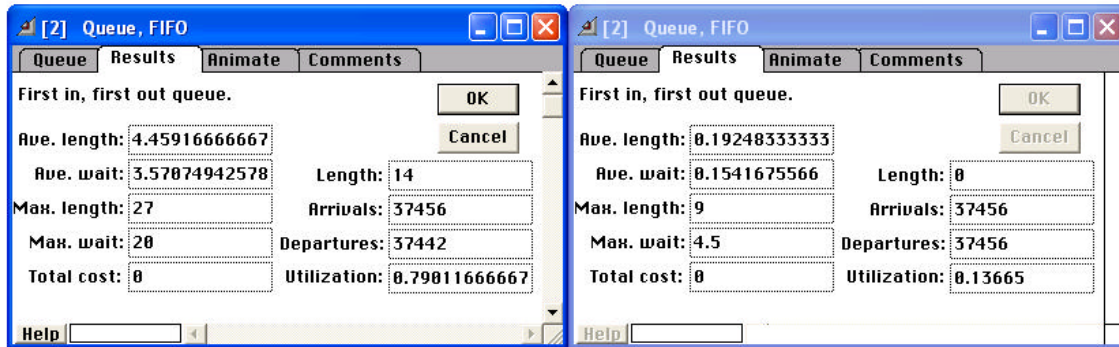


Figura 5.17: Resultados para el modelo antes y después de agregar un Cajero Junior

Hasta este momento sólo se han analizado estadísticas de bloques en particular. Ahora se analizarán estadísticas generales del sistema. Para hacer esto, es necesario modificar el modelo, agregando nuevos bloques y conexiones.

El primer paso consiste en agregar un bloque *Timer* de la librería *Discrete Event*, que calcula el tiempo que los ítems demoran en recorrer la distancia entre dos puntos especificados. Este bloque toma el tiempo desde que un ítem lo atraviesa hasta que llega al bloque que se conecte al sensor. Es por eso que tiene tres conexiones de ítems.

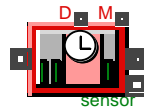


Figura 5.18: Bloque *Timer*

Para utilizar el bloque *Timer* es necesario unir las salidas de los tres cajeros, ya que la entrada de este bloque se conectará al bloque *Generator*, la salida al bloque *Queue, FIFO* y el sensor a la salida de los cajeros. Para unir las salidas de los tres cajeros se utilizarán dos bloques *Combine*, de la librería *Discrete Event*, que combinan ítems de dos fuentes en un solo flujo. El resultado de agregar estos bloques se encuentra en la Figura 5.19.

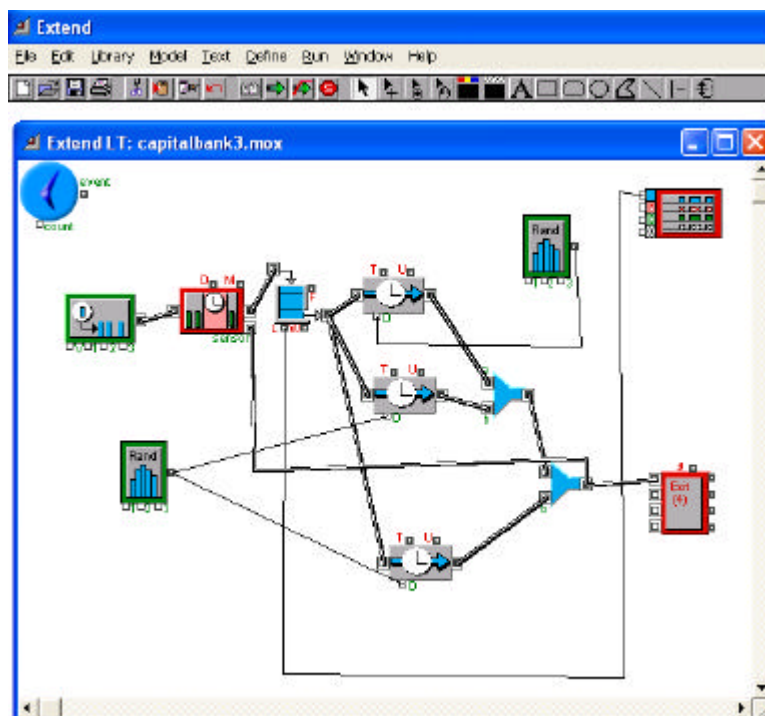


Figura 5.19: Modelo del banco con un bloque *Timer* y dos bloques *Combine*

Al ejecutar la simulación con el bloque *Timer*, se obtiene el tiempo promedio que un cliente pasa en el sistema. Este dato se encuentra en el cuadro de diálogo del bloque, como puede verse en la Figura 5.20. Por lo tanto, el tiempo promedio es de 5.8735 minutos.

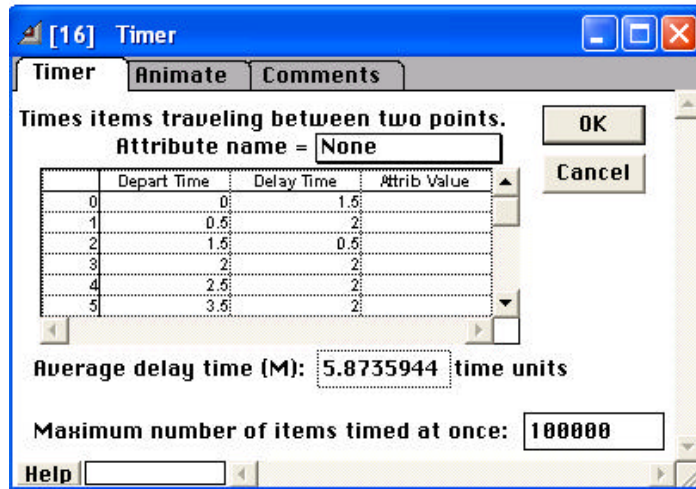


Figura 5.20: Cuadro de diálogo del bloque *Timer*

Conectando el bloque *Timer* a un bloque *Max & Min* se pueden calcular los tiempos mínimo y máximo que un cliente pasa en el sistema. El modelo resultante se encuentra en la Figura 5.21.

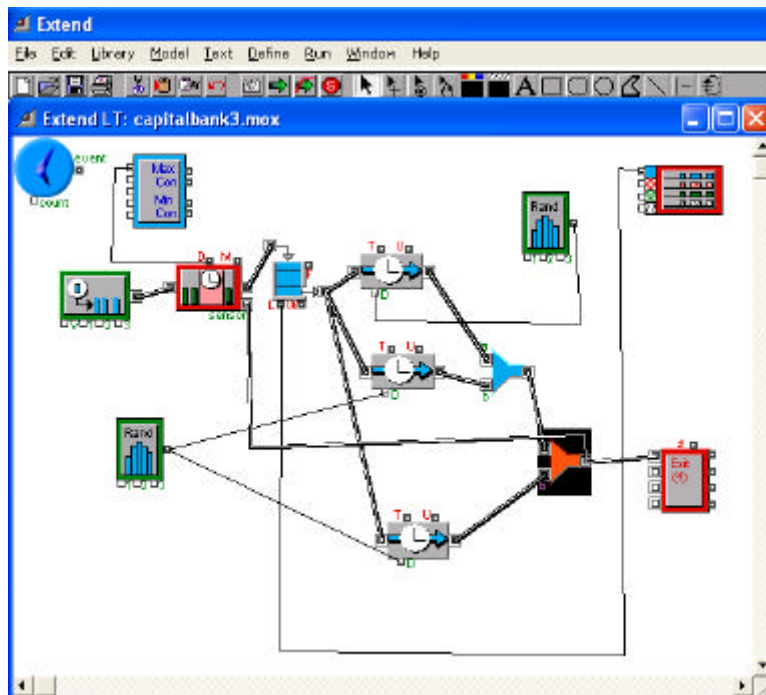


Figura 5.21: Modelo con un bloque *Min & Max*

El cuadro de diálogo del bloque *Min & Max* informa sobre los tiempos máximo y mínimo que un cliente pasa en el sistema. Estos son 23.5 minutos y 0.5 minutos respectivamente.

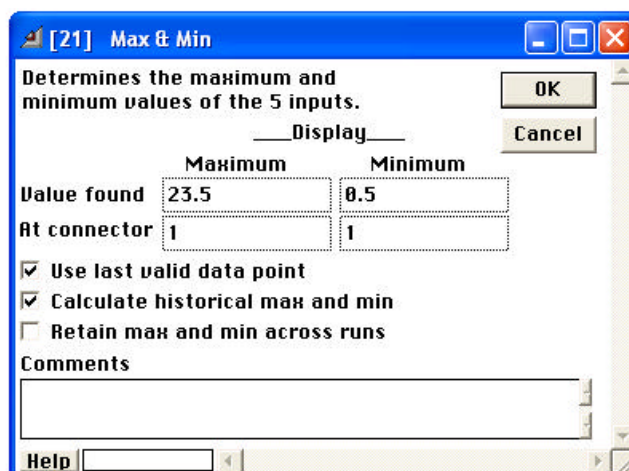


Figura 5.22: Cuadro de diálogo del bloque *Max & Min*

Como se pudo ver en este caso, Extend facilita tanto el desarrollo como el mantenimiento del modelo. Permite probar suposiciones en forma rápida, y realizar análisis de resultados. Extend también evita la realización de tareas que llevan a errores, como la escritura de fórmulas en Excel, ya que la lógica para modelar las situaciones más comunes se encuentra en los bloques contenidos en las librerías.

5.4. Comparación de Resultados

La siguiente tabla compara los resultados obtenidos utilizando Excel y Extend, realizando diez ejecuciones de la simulación en cada caso.

Excel											
Ejecución	1	2	3	4	5	6	7	8	9	10	Promedio
Tiempo promedio en el sistema	5.8	6.5	7.7	4.5	6.8	6.9	7.4	7.5	7.1	6.8	6.7
Espera promedio en la cola	3.5	4.2	5.5	2.2	4.5	4.6	5.2	5.2	4.8	4.5	4.4

Extend											
Ejecución	1	2	3	4	5	6	7	8	9	10	Promedio
Tiempo promedio en el sistema	5.8	6	7.3	5.9	6.1	5.8	6.2	6.6	6.2	5.9	6.2
Espera promedio en la cola	3.5	3.7	5	3.6	3.8	3.5	3.9	4.3	3.9	3.6	3.9

Tabla 5.8: Comparación de resultados en Excel y Extend

Como se puede ver, los resultados son muy similares en ambos casos. El promedio de tiempo en el sistema en Excel es de 6.7 minutos, mientras que en Extend es de 6.2 minutos. El promedio de espera en la cola es de 4.4 minutos en Excel y 3.9 minutos en Extend.

6. Conclusiones

La simulación ha demostrado ser una herramienta de gran utilidad en el campo de la Investigación de Operaciones. Esto se debe principalmente a su flexibilidad, ya que permite resolver problemas cuya solución sería imposible o muy costosa de hallar mediante otras técnicas. Esta característica posibilita la utilización de simulación para sistemas cuya resolución mediante métodos estadísticos no es factible, cuando los mismos no cumplen con las premisas que dichos métodos requieren. Por otro lado, evita incurrir en los costos de desarrollar o modificar un sistema real, permitiendo experimentar con el modelo, para determinar qué sucedería frente a cambios en los valores de los parámetros del sistema.

El surgimiento de los lenguajes de simulación ha permitido que el desarrollo de modelos de simulación sea más rápido y fácil. Su principal ventaja sobre los de propósito general es que incluyen las funciones de modelado más utilizadas. Esto evita que el desarrollador tenga que programar todo el modelo desde cero, pudiendo armarlo a partir de un conjunto de bloques o funciones ya provistas por el lenguaje.

La evolución de los lenguajes de simulación llevó al surgimiento de simuladores; que son programas que permiten desarrollar modelos sin escribir código, mediante la utilización de una interfase gráfica amigable,

bloques, conexiones y parámetros que conforman el modelo. Este ahorro en tiempo de desarrollo obviamente implica una pérdida de flexibilidad, ya que si un sistema no puede modelarse con los bloques incluidos en el simulador, éste no puede ser simulado.

Un híbrido entre lenguaje de simulación y simulador es Extend. Esta característica le permite proveer tanto la facilidad de modelado de un simulador, como la flexibilidad de un lenguaje de simulación. La gran cantidad de bloques incluidos en Extend hace que rara vez sea necesario programar un bloque, lo cual permite desarrollar modelos en forma intuitiva, inclusive por personas sin conocimientos de programación. Si el sistema que se va a desarrollar requiere bloques no incluidos en el paquete, se pueden desarrollar utilizando ModL, el lenguaje de simulación incluido en Extend. Este lenguaje es muy similar a C, por lo cual es accesible a cualquier persona con conocimientos de programación en C.

El caso práctico del banco resuelto en el Capítulo 5 permite advertir la facilidad de uso y el poder de Extend. Mediante la utilización de iconos, cuadros de diálogo, bloques, drag & drop y conexiones se creó un modelo que simula la cola de un banco. La comparación del desarrollo de esta misma simulación en forma manual y en Excel permite apreciar las ventajas de Extend. Algunas de ellas son la velocidad con que se desarrolla un modelo, la facilidad de modelado, el evitar tareas que generen errores y la facilidad para el mantenimiento. Esto fue ilustrado agregando un cajero más al banco, lo cual, en Extend, puede hacerse en pocos minutos, simplemente agregando el bloque nuevo y conectándolo. Si se quisiera hacer esto en Excel, se deberían modificar prácticamente todas las fórmulas, lo cual sería una tarea tediosa, que llevaría más tiempo y podría generar errores. En forma manual se debería realizar todo el trabajo desde cero para introducir esa modificación.

Existen varias posibilidades para realizar extensiones a esta tesina. Una de ellas consistiría en tomar otro lenguaje de simulación, como por ejemplo GPSS, ProModel, Simul8 o Crystal Ball, analizarlo y desarrollar el modelo del caso práctico en ese lenguaje, para luego realizar una comparación con Extend.

Otra posibilidad sería utilizar Extend para simular un sistema continuo, focalizando la tesina en la simulación de este tipo de sistemas, y obviando lo referido a la simulación de eventos discretos.

Una última opción que se plantea es la de desarrollar, en Extend, una librería de bloques que puedan ser utilizados en un campo específico. En este caso, el punto más importante estaría dado por el lenguaje de programación ModL, ya que mediante éste se programarían los bloques integrantes de la librería, que luego podrían utilizarse para el desarrollo de modelos en forma gráfica.

7. Anexo A: Tabla de Números Pseudo-Aleatorios

La siguiente tabla de números pseudo-aleatorios fue extraída del apéndice C del libro «Applied Management Science» de Lawrence y Pasternack.

8. Anexo B: Tabla de Lenguajes de Simulación

La siguiente tabla enumera los principales lenguajes de simulación y algunas de sus características.

Software	Compañía	Aplicaciones Típicas	Mercados Principales	Requerimientos de sistema	
				RAM	Sistema operativo
Analytica	Lumina Decision Systems, Inc.	Modelado de negocios, análisis de decisiones y riesgo, utilizado como una alternativa más flexible a las hojas de cálculo	Industria, gubernamental, educación, telecomunicaciones, salud, farmacéutica, energía, ambiental, manufactura, automotriz, transporte, defensa	8 MB	Windows 95, 98, NT, 2000, XP y ME; Mac OS 7, 8 y 9
AnyLogic 5.0	XJ Technologies	Dinámica de sistemas y negocios, análisis de performance, costos y riesgos, optimización, planificación, soporte a las decisiones	Dirección estratégica, manufactura, servicios, logística, supply chain, manejo de materiales, salud, transporte, IT, telecomunicaciones, científica	128MB min., 512MB recom.	Windows NT, 2000 y XP
Arena	Rockwell Software	Manufactura, supply chain, procesos de negocios, salud, militar, mejoras de almacenamiento y logística	Manufactura, supply chain, logística, procesos de negocios, militar, salud	64MB min., 128MB recom.	Windows 95, 98, ME, NT, 2000 y XP
AutoMod	Brooks Automation	Manejo de materiales, almacenamiento, manejo de equipaje, manufactura	Automotriz, aeroespacial, operaciones de aeropuertos, manufactura, almacenamiento, distribución	512MB o más recom.	Windows OS
Crystal Ball Professional Edition	Decisioneering, Inc.	Planeamiento y análisis de negocios, análisis de costo/beneficio, manejo de riesgos, exploración de petróleo, optimización de portfollio, administración de proyectos	Servicios financieros, planeamiento financiero, petrolera, farmacéutica, telecomunicaciones, manufactura, energía, seguros, gubernamental, aeroespacial	16MB min.	Windows 95, 98, Me, 2000, NT 4 y XP
Crystal Ball Standard Edition	Decisioneering, Inc.	Planeamiento y análisis de negocios, análisis de costo/beneficio, manejo de riesgos, exploración de petróleo, evaluación ambiental, administración de proyectos	Servicios financieros, planeamiento financiero, petrolera, farmacéutica, telecomunicaciones, manufactura, energía, seguros, gubernamental, aeroespacial	16MB min.	Windows 95, 98, Me, 2000, NT 4 y XP
DecisionPro	Vanguard Software Corporation	Modelado de negocios financieros, optimización de procesos, toma de decisiones	Dirección, consultoría, petrolera, manufactura, legal	16MB	Windows 95, 98, 2000 , NT4 y XP
DecisionScript	Vanguard Software Corporation	Planeamiento financiero basado en la Web, asistencia online de ventas, reportes, simulación de portfollio	Servicios financieros, desarrollo de sitios Web	Cliente: 32MB; Servidor: 128MB	Windows 95, 98, 2000 , NT4 y XP
eM-Plant	Tecnomatix Technologies Inc.	Manufactura, manejo de materiales, simulación de procesos de negocios, logística, distribución, planeamiento, verificación de procesos, supply chain	Manufactura discreta (automotriz, electrónica, etc.), logística, distribución, consultoría, salud, bancaria	64MB min., 512MB recom.	Windows NT, 2000 y XP
Enterprise Dynamics 5.0	Incontrol Enterprise Dynamics	Simulación, emulación, análisis de capacidad, manejo de materiales, diseño de instalaciones, utilización de recursos, supply chain	Paquetes de simulación dedicados a manufactura, manejo de materiales, aeropuertos, investigación y educación, industria del acero, puertos	64MB	Windows 95, 98, 2000, NT, ME y XP
ExpertFit	Averill M. Law & Associates	Ajusta distribuciones de probabilidad a datos en forma automática y precisa, ayuda a elegir distribuciones en caso de ausencia de datos	Manufactura, defensa, comunicaciones, transporte, salud, reingeniería de procesos, call centers, servicios, análisis de datos en general	64MB	Windows 98, ME, 2000, NT y XP

Software	Compañía	Aplicaciones Típicas	Mercados Principales	Requerimientos de sistema	
				RAM	Sistema operativo
Extend Industry	Imagine That, Inc.	Indispensable para proyectos de simulación de gran escala. Incluye una base de datos relacional interna y un módulo para simular	Sistemas de gran tamaño y basados en ratios, incluyendo logística, call centers de gran volumen, líneas de empaque, etc.	64MB min., 128MB+ recom.	Windows XP, 2000, NT, ME, 98; Mac OS X y 9.1+
Extend OR	Imagine That, Inc.	Arquitectura de eventos discretos para modelar cualquier proceso que involucre objetos físicos o lógicos en movimientos	Modelado de manufactura y negocios, sistemas de comunicación, salud, transporte, supply chain, etc.	64MB min., 128MB+ recom.	Windows XP, 2000, NT, ME, 98; Mac OS X y 9.1+
Extend Suite	Imagine That, Inc.	Modela procesos continuos, de eventos discretos, o de ratios, además de incluir un sistema de administración de datos relacional	Sistemas de gran escalas, manufactura, logística, líneas de empaque, transporte, negocios, call centers, ingeniería, científica	64MB min., 128MB+ recom.	Windows XP, 2000, NT, ME, 98; Mac OS X y 9.1+
Factory Explorer	Wright Williams & Kelly	Optimización de márgenes, reducción del tiempo de ciclo, planeamiento, análisis de cuellos de botella	Semiconductores, monitores de pantalla plana, paneles solares, automotriz, manufactura discreta	256MB min., 512MB recom.	Windows 98 o superior
FirstSTEP Designer	Interfacing Technologies	FirstSTEP Designer es una herramienta de administración de procesos de negocios que le brinda información a los que toman decisiones	Empresas que quieren establecer estándares de operación o implementar procesos regulados	32MB	Windows 95, 98, 2000, NT y XP
Flexsim	Flexsim Software Products, Inc	Manufactura, manejo de materiales, almacenamiento, distribución, monitoreo en tiempo real, supply chain, envío de containers	Manufactura, manejo de materiales, almacenamiento, distribución, monitoreo en tiempo real, supply chain, envío de containers	128MB min., 256MB recom.	Windows XP y 2000
GAUSS	Aptech Systems, Inc.	GAUSS es un lenguaje de programación científico y estadístico con múltiples herramientas	Económica, financiera, ciencias sociales, ingeniería, y cualquier área que realice análisis de datos o ajuste de modelos	256MB recom.	Windows, AIX4, HPUNIX11, Solaris y Linux
GoldSim	GoldSim Technology Group	Planeamiento estratégico, análisis y administración de riesgos, dinámica de negocios, modelado de sistemas de ingeniería, administración de portfolio, modelado ambiental	Manufactura, minería, recursos de agua, seguros, energía, gubernamental	64MB	Windows 98 o superior
GPSS World for Windows	Minuteman Software	Modelado de manufactura, telecomunicaciones, redes de computadoras, redes de colas, modelado de sistemas de eventos discretos	Departamento de investigación de operaciones, profesionales de ingeniería industrial realizando modelado de simulaciones	32MB	Windows 98, ME, 2000, NT y XP
HighMAST™	Highpoint Software Systems, LLC	Framework basado en .Net para desarrollo de componentes y servicios de simulación para ser integrados en sistemas más grandes	Varios. Actualmente en manufactura y validación de planeamiento	160 MB; 256MB	Servidor: Windows NT o superior, con .Net Framework; Cliente: Varios
MAST	CMS Research Inc.	Diseño y planeamiento de sistemas flexibles de trabajo y máquinas	Manufactura que utilice la flexibilidad para sustituir el inventario	128MB	Windows 2000, XP, NT o superior
MedModel	ProModel Solutions	Tiempos de espera, flujo de pacientes, administración de camas, locación de espacio y recursos, diseño de instalaciones, análisis de costos	Salud	64MB min., 128MB recom.	Windows 98 o superior
Micro Saint	Micro Analysis & Design, Inc.	Se utiliza principalmente para modelar aplicaciones en defensa, manufactura y servicios	Defensa, manufactura	64MB	Windows 95, 98, 2000, ME, XP y NT
mystrategy	Global Strategy Dynamics Ltd.	Estrategia y planeamiento de negocios, modelado de arquitectura de negocios	Cualquier sistema de negocios, los usuarios son planeadores estratégicos y management en general	32MB min., 64MB+ recom.	Windows 95 o superior

				Requerimientos de sistema	
Software	Compañía	Aplicaciones Típicas	Mercados Principales	RAM	Sistema operativo
NAG C Library	Numerical Algorithms Group	Logística, portfolio/producto, administración de riesgos, planeamiento, programación lineal, optimización, programación no lineal	Industria química, financiera, militar, manufactura, salud, petrolera, ambiental, biotecnológica, telecomunicaciones, transporte	256MB o más	Windows, Linux, Unix, etc.
NAG SMP Library	Numerical Algorithms Group	Logística, portfolio/producto, administración de riesgos, planeamiento, programación lineal, optimización, programación no lineal	Industria química, financiera, militar, manufactura, salud, petrolera, ambiental, biotecnológica, telecomunicaciones, transporte	256MB o más	Windows, Linux, Unix, etc.
PASION Simulation System	Stanislaw Raczynski	Soporta eventos discretos, colas, modelos continuos, gráficos, dinámica de cuerpos rígidos	Educación, universidades, consultoras de simulación y empresas en general	64Mb min.	Windows 98 o superior
PIMSS	MJC2 Limited	Modelado, optimización y planificación de operaciones de manufactura	Comidas, bebidas, petróleo, cemento, químicos, farmacéuticos, automotrices, textiles, muebles, construcción, plásticos, telecomunicaciones	256MB min., 512MB recom.	Windows 2000/XP, UNIX y Linux
ProcessModel	ProcessModel, Inc.	Análisis de procesos de negocios, mejora de procesos, certificación ISO	Puede analizar y mejorar procesos de eventos discretos en todos los mercados	64MB min., 128MB recom.	Windows 95, NT, 98, 2000 y XP
ProModel	ProModel Solutions	Planeamiento de proyectos y portfolio, análisis de costos, mejora de procesos, supply chain	Manufactura y logística, farmacéutica	64MB min., 128MB recom.	Windows 98 o superior
Proplanner Manufacturing Process Management Software	Proplanner	Planeamiento e ingeniería de procesos para crear, documentar, administrar y acceder a la información de los procesos de manufactura	Manufactura	32MB min., 96MB+ recom.	-
ProVision	Proforma Corporation	Usado principalmente para mejora y rediseño de procesos, modelado de sistemas, ISO/TQM, arquitectura empresarial	Empresas que forman parte de Fortune 1000 y consultoras con foco en la manufactura, telecomunicaciones, finanzas, ventas y gubernamentales	256MB	Windows 98 o superior
Resource Manager	User Solutions, Inc.	Modelos de planeamiento de manufactura con pruebas de alternativas de inventario y logística	Management de manufactura y operaciones, con consideraciones de gente, máquinas y materiales	128MB min., 512MB recom.	Windows 98, 2000, NT, XP, etc.
SAIL	CMS Research Inc.	Planeamiento diario y monitoreo de performance de sistemas de manufactura	Manufactura	128MB	Windows NT, 2000, XP o superior
SansGUI Modeling and Simulation Environment	ProtoDesign Inc.	Construcción de modelos para ciencia e ingeniería, desarrollo de programas de simulación, gráficos y visualizaciones dinámicas	Sistemas generales, investigación científica, ingeniería, desarrollo de software	64MB	Windows 95/98, ME, NT4, 2000 y XP
SAS Software	SAS Institute Inc.	Tráfico de redes, actividad de call center, tráfico de sala de emergencias, actividad de redes de computadoras, niveles de inventario, modelos de cola	Sector público, petrolera, telecomunicaciones, salud, ventas, bancaria, académica	Varía	Los principales Unix y Windows
ServiceModel	ProModel Solutions	Mejora y optimización de procesos, análisis de capacidad, de transacciones y financiero, niveles de personal, planeamiento, transporte, sistemas de seguridad	Financiera, bancaria, seguros, aerolíneas, call centers, y otras industrias de servicios	64MB min., 128MB recom.	Windows 98 o superior
ShowFlow	Webb Systems Limited	Planeamiento de instalaciones nuevas, reducción de tiempo de ciclo, análisis de cuellos de botella, análisis de niveles de personal	Manufactura, logística, servicios financieros, operaciones de ventas	64MB+	Windows 9x, NT4, Me y XP

Software	Compañía	Aplicaciones Típicas	Mercados Principales	Requerimientos de sistema	
				RAM	Sistema operativo
SIGMA	Custom Simulations	Sistemas de eventos discretos	Sistemas de manufactura y servicios, entre ellos empresas de bioproducción, semiconductores, salud y bancos	-	Todas las versiones de Windows
SimCAD Pro	CreateASoft, Inc.	Manufactura, línea de montaje, robótica, automatización de laboratorios, diseño de fábricas	Manufactura, línea de montaje, robótica, automatización de laboratorios, diseño de fábricas	128MB	Windows NT, 2000 y XP
SIMPROCESS	CACI Products Company	SIMPROCESS es un componente clave en el proceso de negocios y ciclo de vida de la reingeniería	SIMPROCESS fue diseñado para organizaciones que requieren analizar escenarios y mitigar riesgos	256MB min., 512MB recom.	Windows
SIMUL8 Professional	SIMUL8 Corporation	Análisis de performance, eliminación de cuellos de botella, desarrollo de nuevos productos y procesos, análisis de capacidad, mejora continua, escenarios de pruebas	Procesos de negocios, call centers, manufactura, supply chain, logística, salud, financiera, farmacéutica, servicio al cliente	64MB	Windows 95 o superior
SIMUL8 Standard	SIMUL8 Corporation	Análisis de performance, eliminación de cuellos de botella, desarrollo de nuevos productos y procesos, análisis de capacidad, mejora continua, escenarios de pruebas	Procesos de negocios, call centers, manufactura, supply chain, logística, salud, financiera, farmacéutica, servicio al cliente	-	Windows 95 o superior
SLIM	MJC2 Limited	Modelado estratégico, optimización y análisis de redes de distribución, operaciones de transporte y redes de supply chain	Logística, distribución, petrolera, ventas, transporte, química, construcción, comidas, bebidas	256MB min., 512MB recom.	Windows 2000/XP, UNIX, Linux
Supply Chain Builder	Simulation Dynamics	Se usa para entender complejos sistemas de supply chain y distribución	Automotriz, bienes de consumo, comidas y bebidas, transporte, farmacéutica	128MB min., 1GB recom.	Windows 2000/XP
VisSim	Visual Solutions	Sistemas no lineales, diseño de control, sistemas embebidos, logística, economía, simulación en tiempo real de plantas virtuales	Automotriz, aeroespacial, energía, proceso, control industrial, planeamiento de logística	-	Windows 9x, ME, NT, 2000, XP, y Solaris
Visual Simulation Environment	Orca Computer, Inc.	El software Visual Simulation Environment es un ambiente integrado de desarrollo y ejecución de simulaciones de eventos discretos	El software Visual Simulation Environment es un ambiente integrado de desarrollo y ejecución de simulaciones de eventos discretos	128MB min.	Windows 98, NT, 2000 y XP

Desarrollo de modelos									
Software	Desarrollo de modelos en forma gráfica	Desarrollo de modelos mediante programación	Depuración en tiempo de ejecución	Ajuste a distribuciones de probabilidad (Especificar)	Soporte de análisis de datos de salida (Especificar)	Ejecuciones por lotes o diseño de experimentos (Especificar)			
Analytica	Sí	Sí	Sí	Sí	Métodos simples	Sí	Análisis gráfico, análisis de sensibilidad, y muchas funciones estadísticas	Sí	Muestreo aleatorio o sistemático para selección de entradas
AnyLogic 5.0	Sí	Sí	Sí	Sí	Stat::Fit soporta más de 40 distribuciones AnyLogic	Sí	Recolección de datos y procesamiento estadístico (media, desviación, etc.), presentación (Gantt, gráficos, histogramas)	Sí	Soporta simulación, optimización, Monte Carlo, análisis de sensibilidad, experimentos personalizados basados en scripts
Arena	Sí	Sí	Sí	Sí	-	Sí	-	Sí	-
AutoMod	Sí	Sí	Sí	Sí	Utilizando ExpertFit	Sí	El módulo AutoStat provee análisis estadístico durante la fase de experimentación del proyecto	Sí	Ejecuciones por lotes con AutoStat; AutoMod permite diseño de experimentos
Crystal Ball Professional Edition	No	No	No	Sí	Tres métodos de ajuste (Chi-Square, Kolmogorov-Smirnov, Anderson-Darling) para distribuciones continuas	Sí	Analiza estadísticas y muestra los resultados gráficamente. Puede extraer datos y crear reportes	Sí	El Developer Kit contiene macros y funciones que pueden ser programados para automatizar toda la simulación
Crystal Ball Standard Edition	No	No	No	Sí	Tres métodos de ajuste (Chi-Square, Kolmogorov-Smirnov, Anderson-Darling) para distribuciones continuas	Sí	Analiza estadísticas y muestra los resultados gráficamente. Puede extraer datos y crear reportes	No	-
DecisionPro	Sí	Sí	Sí	Sí	Uniforme, discreta, triangular, normal, logarítmica, gama, exponencial, beta, Bernoulli, binomial, Poisson, personalizada	Sí	Estadísticas de simulación, análisis de sensibilidad, presentaciones gráficas	Sí	Ejecuciones por lotes o interactivas
DecisionScript	Sí	Sí	Sí	No	-	Sí	Estadísticas de simulación, análisis de sensibilidad, presentaciones gráficas, todas disponibles como páginas Web	Sí	Ejecuciones por lotes o interactivas
eM-Plant	Sí	Sí	Sí	Sí	Herramientas de análisis de datos estándar incluidas (DataFit)	Sí	Herramientas de análisis de datos estándar incluidas (DataFit)	Sí	Manejador de experimentos, soporta ejecuciones por lotes, diseño de experimentos por reglas, redes neuronales, cálculo de intervalos de confianza

Desarrollo de modelos									
Software	Desarrollo de modelos en forma gráfica	Desarrollo de modelos mediante programación	Depuración en tiempo de ejecución	Ajuste a distribuciones de probabilidad (Especificar)	Soporte de análisis de datos de salida (Especificar)	Ejecuciones por lotes o diseño de experimentos (Especificar)			
Enterprise Dynamics 5.0	Sí	Sí	Sí	Sí	Ajuste a curvas incluido	Sí	Incluye reportes, gráficos, conexión a Excel, conexiones DDE, IP, ActiveX y ODBC para software de análisis externo	Sí	Diseño de experimentos, ejecución por lotes mediante scripts
ExpertFit	-	-	-	Sí	40 distribuciones, 4 pruebas de exactitud del ajuste	-	-	Sí	Varios conjuntos de datos pueden ser analizados simultáneamente
Extend Industry	Sí	Sí	Sí	Sí	-	Sí	Se pueden calcular intervalos de confianza haciendo click en un botón	Sí	Soporta ejecución automatizada de diferentes escenarios
Extend OR	Sí	Sí	Sí	Sí	-	Sí	Se pueden calcular intervalos de confianza haciendo click en un botón	Sí	Soporta ejecución automatizada de diferentes escenarios
Extend Suite	Sí	Sí	Sí	Sí	Incluye Stat::Fit	Sí	Se pueden calcular intervalos de confianza haciendo click en un botón	Sí	Soporta ejecución automatizada de diferentes escenarios
Factory Explorer	No	Sí	Sí	No	-	Sí	Intervalos de confianza, prueba de modificación por inicialización	Sí	Se puede almacenar una secuencia de ejecución por lotes en una hoja de cálculo de Excel
FirstSTEP Designer	Sí	Sí	Sí	Sí	Se ofrecen varias distribuciones diferentes. No soporta ajuste de datos	Sí	Análisis y reporte de procesos para planeamiento de recursos, ABC, camino crítico y tiempo de ciclo	No	-
Flexsim	Sí	Sí	Sí	Sí	Flexsim soporta 29 distribuciones diferentes. Flexsim incluye el software de ajuste de distribuciones ExpertFit	Sí	Flexsim soporta 29 distribuciones diferentes. Flexsim incluye el software de ajuste de distribuciones ExpertFit	Sí	Flexsim incluye un módulo de diseño de experimentos que permite definir las ejecuciones por lotes de un modelo
GAUSS	No	Sí	Sí	Sí	Amplio rango de modelos estadísticos disponibles	Sí	Intervalos de confianza, perfil t, y otras métricas	Sí	Modo por lotes o interactivo
GoldSim	Sí	Sí	No	No	-	Sí	Gráficos, análisis de correlación, estadísticas de distribución, exportación de resultados	No	-
GPSS World for Windows	Sí	Sí	Sí	No	-	Sí	Análisis de varianza	Sí	Diseño y ejecución automática de experimentos de optimización
HighMAST™	No	Sí	Sí	Sí	El framework permite generar distribuciones que varían en el tiempo	Sí	Salidas en histogramas y series de datos.	Sí	Ejemplo de demostración que realiza análisis de Monte Carlo
MAST	Sí	Sí	Sí	No	-	Sí	Permite comparar resultados simulados para determinar problemas de capacidad o integración	No	-

Desarrollo de modelos										
Software	Desarrollo de modelos en forma gráfica	Desarrollo de modelos mediante programación	Depuración en tiempo de ejecución	Ajuste a distribuciones de probabilidad (Especificar)	Soporte de análisis de datos de salida (Especificar)	Ejecuciones por lotes o diseño de experimentos (Especificar)				
MedModel	Sí	Sí	Sí	Sí	Distribuciones definidas por el usuario, 15 distribuciones predefinidas, más ajuste de distribuciones usando Stat::Fit (incluido)	Sí	Reportes y gráficos completos incluidos, exporta a Excel y Access para análisis	Sí	Se pueden definir escenarios ilimitados para experimentar con los parámetros	
Micro Saint	Sí	Sí	Sí	No	-	Sí	Algunos gráficos y estadísticas automáticas están disponibles, los resultados pueden ser exportados para análisis	No	-	
mystrategy	Sí	No	No	No	-	No	-	No	-	
NAG C Library	No	Sí	Sí	No	-	No	-	Sí	Por lotes	
NAG SMP Library	No	Sí	Sí	No	-	No	-	Sí	Por lotes	
PASION Simulation System	Sí	Sí	No	Sí	Uniforme, normal, logarítmica, triangular, exponencial, Erlang, Beta, Gama, Weibull, ejemplo (empírica) y personalizada	Sí	Varias representaciones de resultados gráficos, entre ellos varianza e intervalos de confianza en función del tiempo, modelos de colas	Sí	Simulaciones que se repiten automáticamente para análisis de varianza	
PIMSS	Sí	Sí	Sí	Sí	Herramientas configurables para permitir mapeos de variables de entrada	Sí	La salida es gráfica y/o tabular	-	-	
ProcessModel	Sí	Sí	Sí	Sí	Incluye el software de ajuste de curvas Stat::Fit	Sí	Análisis general y detallado	Sí	El planificador de escenarios permite realizar experimentos por lotes	
ProModel	Sí	Sí	Sí	Sí	Distribuciones definidas por el usuario, 15 distribuciones predefinidas, más el software Stat::Fit incluido	Sí	Reportes de análisis de salida, exporta a Excel y Access para análisis	Sí	Pueden definirse escenarios ilimitados para experimentar parámetros	
Proplanner Manufacturing Process Management Software	Sí	No	No	No	-	Sí	Se administran y comparten los datos, se exportan como archivos de texto	No	-	
ProVision	Sí	No	-	Sí	ProVision provee 11 distribuciones diferentes para tiempos de arribo y de procesamiento	Sí	Gráficos y tablas para costos, tiempos y uso de recursos como resultado de una ejecución de la simulación	Sí	El simulador de ProVision se puede llamar desde otra aplicación con una interfase COM	
Resource Manager	No	Sí	Sí	No	-	Sí	Exportación de datos a Excel con consultas y reportes personalizados	Sí	Se pueden hacer ejecuciones en modo Planned	

Desarrollo de modelos									
Software	Desarrollo de modelos en forma gráfica	Desarrollo de modelos mediante programación	Depuración en tiempo de ejecución	Ajuste a distribuciones de probabilidad (Especificar)	Soporte de análisis de datos de salida (Especificar)	Ejecuciones por lotes o diseño de experimentos (Especificar)			
SAIL	Sí	Sí	Sí	No	-	Sí	Optimiza la asignación de capacidad usando criterios de performance	Sí	Usa métodos de búsqueda heurísticos de 4 variables de entrada para definir las asignaciones de capacidad
SansGUI Modeling and Simulation Environment	Sí	Sí	Sí	No	-	Sí	Usa código en C, C++ o Fortran, permite vistas de resultados de simulación, gráficos dinámicos y en 3D	Sí	Parámetros para varias ejecuciones, permite ejecuciones externas mediante scripts
SAS Software	Sí	No	Sí	Sí	Capacidades estadísticas amplias en SAS	Sí	Varios análisis incluidos además de análisis estadísticos externos	No	-
ServiceModel	Sí	Sí	Sí	Sí	Distribuciones definidas por el usuario, 15 distribuciones predefinidas, más el software Stat:Fit incluido	Sí	Reportes de análisis de salida, exporta a Excel y Access para análisis	Sí	Pueden definirse escenarios ilimitados para experimentar parámetros
ShowFlow	Sí	Sí	Sí	Sí	Tiene una herramienta de ajuste que busca la mejor distribución para las entradas	Sí	Reportes y gráficos definidos por el usuario, exporta a Excel	Sí	Soporta tanto ejecuciones por lotes como diseño de experimentos
SIGMA	Sí	Sí	Sí	Sí	Soporta la mayoría de las distribuciones de entrada	Sí	Gráficos y estadísticas	Sí	Permite experimentación mediante hoja de cálculo o por Web
SimCAD Pro	Sí	Sí	Sí	Sí	-	Sí	-	Sí	-
SIMPROCESS	Sí	Sí	No	Sí	SIMPROCESS incluye el software de ajuste de distribuciones ExpertFit	Sí	Los reportes pueden ser exportados a bases de datos u hojas de cálculo	Sí	SIMPROCESS incluye un administrador de experimentos para ejecutar múltiples escenarios de simulación
SIMUL8 Professional	Sí	Sí	Sí	Sí	Stat:Fit incluido con el software	Sí	Reportes estadísticos personalizados incluyendo gráficos e intervalos de confianza, exporta a Excel	Sí	-
SIMUL8 Standard	Sí	Sí	Sí	Sí	Stat:Fit se puede agregar por \$245	Sí	Reportes estadísticos personalizados incluyendo gráficos e intervalos de confianza, exporta a Excel	Sí	-
SLIM	Sí	Sí	Sí	Sí	Herramientas configurables para permitir mapeos de variables de entrada	Sí	La salida es gráfica y/o tabular	-	-
Supply Chain Builder	Sí	Sí	Sí	Sí	Mediante una base de datos interna	Sí	Análisis de reportes generados por el modelo	Sí	Permite ejecutar una serie de experimentos diseñados
VisSim	Sí	Sí	Sí	Sí	Mediante un optimizador incluido	Sí	Análisis on-line de reportes	Sí	Soporta simulación de Monte Carlo
Visual Simulation Environment	Sí	Sí	Sí	No	-	Sí	Provee una herramienta para analizar reportes de salida		

Desarrollo de modelos (continuación)						
Software	Optimización (Especificar)	Reusabilidad de código	Ejecución de modelos (¿Puede verse un modelo completo sin el software para desarrollo?)	Herramientas para ejecutar modelos	¿Estas herramientas tienen un costo?	
Analytica	Sí Métodos con parámetros y automáticos	Sí	Sí	Versión de ejecución gratuita, o mediante el sitio Web	No	
AnyLogic 5.0	Sí Incluye el motor OptQuest puede manejar optimizaciones de problemas clásicos y de gran escala	Sí	Sí	No requiere herramientas especiales	No	
Arena	Sí -	Sí	Sí	-	No	
AutoMod	Sí La optimización está basada en el algoritmo de estrategias de evolución	Sí	Sí	Licencia para ejecución (con costo) o AutoView Player (gratis)	-	
Crystal Ball Professional Edition	Sí Incluye el motor OptQuest puede manejar optimizaciones incluso con incertidumbre	No	Sí	Los modelos en Excel que fueron mejorados con Crystal Ball pueden ser vistos por cualquier usuario de Excel	No	
Crystal Ball Standard Edition	No -	No	Sí	Los modelos en Excel que fueron mejorados con Crystal Ball pueden ser vistos por cualquier usuario de Excel	No	
DecisionPro	Sí Simplex, integer	Sí	Sí	Se pueden publicar modelos como páginas Web mediante DecisionScript Server	Sí	
DecisionScript	Sí Simplex, integer	Sí	Sí	DecisionScript Server	Sí	
eM-Plant	Sí Algoritmos genéticos, redes neuronales	Sí	Sí	Diferentes niveles de empaquetado: Librerías (fácil modelado), Modelo (cambios de parámetros), Visualizados (sólo simulación, sin costo)	Sí	
Enterprise Dynamics 5.0	Sí A través de OptQuest e Issop	Sí	Sí	Versión visualizador (gratis) y versión de ejecución (con costo)	-	
ExpertFit	- -	-	-	-	-	
Extend Industry	Sí Todas las versiones de Extend incluyen un optimizador evolutivo	Sí	Sí	Una versión gratuita se puede bajar del sitio Web para abrir, ver y ejecutar modelos	No	
Extend OR	Sí Todas las versiones de Extend incluyen un optimizador evolutivo	Sí	Sí	Una versión gratuita se puede bajar del sitio Web para abrir, ver y ejecutar modelos	No	

Desarrollo de modelos (continuación)					
Software	Optimización (Especificar)	Reusabilidad de código	Ejecución de modelos (¿Puede verse un modelo completo sin el software para desarrollo?)	Herramientas para ejecutar modelos	¿Estas herramientas tienen un costo?
Extend Suite	Sí Todas las versiones de Extend incluyen un optimizador evolutivo	Sí	Sí	Una versión gratuita se puede bajar del sitio Web para abrir, ver y ejecutar modelos	No
Factory Explorer	No -	Sí	Sí	Las condiciones del modelo, los reportes y los gráficos se almacenan en hojas de cálculo de Excel y pueden ser compartidos	No
FirstSTEP Designer	Sí Se dividen los procesos en eventos discretos fácilmente, se selecciona el mejor proceso realizando análisis	Sí	Sí	XML, HTML, CSV	No
Flexsim	Sí Flexsim incluye un módulo de diseño de experimentos que permite definir las ejecuciones por lotes de un modelo	Sí	Sí	El módulo Flexsim's Presenter es autónomo, no requiere software externo	-
GAUSS	Sí Programación no lineal, programación cuadrática, programación lineal, optimización sin restricciones	Sí	Sí	Módulo GAUSS Run-Time para ejecutar modelos	Sí
GoldSim	No -	Sí	Sí	Está disponible una versión gratuita que permite ver, modificar y ejecutar modelos	No
GPSS World for Windows	Sí Se pueden diseñar y ejecutar experimentos automáticamente	Sí	-		-
HighMAST™	No -	Sí	Sí	Visual Studio.Net, o cualquier compilador que acepte C#	Sí
MAST	Sí MAST itera sobre varios niveles de inventario para determinar el de mayor performance	-	No	-	Sí
MedModel	Sí Optimización disponible mediante OptQuest o SimRunner	Sí	Sí	Los modelos pueden verse con el ProModel Player, que es gratuito	No
Micro Saint	Sí OptQuest busca valores óptimos para las variables en forma automática	No	Sí	Sólo Micro Saint	No
mystrategy	No -	No	No	-	No
NAG C Library	Sí Lineal, no lineal, con restricciones, sin restricciones, cuadrática, suma de cuadrados	Sí	Sí	El modelo puede incorporarse a aplicaciones existentes en C, Java, C#, C++, VB, Fortran, etc.	No

Desarrollo de modelos (continuación)						
Software	Optimización (Especificar)	Reusabilidad de código	Ejecución de modelos (¿Puede verse un modelo completo sin el software para desarrollo?)	Herramientas para ejecutar modelos	¿Estas herramientas tienen un costo?	
NAG SMP Library	Sí Lineal, no lineal, con restricciones, sin restricciones, cuadrática, suma de cuadrados	Sí	Sí	El modelo puede incorporarse a aplicaciones existentes en C, Java, C#, C++, VB, Fortran, etc.	No	
PASION Simulation System	No -	Sí	No	-	-	
PIMSS	Sí PIMSS puede optimizar complejos sistemas de producción y manufactura	Sí	No	-	-	
ProcessModel	Sí Software de optimización SimRunner incluido	Sí	Sí	Por Web	No	
ProModel	Sí Optimización a través del software OptQuest y/o SimRunner	Sí	Sí	Los modelos pueden verse con la versión ProModel Player, que es gratuita	No	
Proplanner Manufacturing Process Management Software	Sí El módulo Line Balancing provee optimización	Sí	No	-	No	
ProVision	Sí Mediante experimentación del usuario	Sí	Sí	Por Web	No	
Resource Manager	Sí Optimización de tiempos de ciclo	Sí	Sí	Excel	Sí	
SAIL	Sí Continúa la búsqueda hasta encontrar 3 de las variables de performance	No	Sí	Versión de ejecución gratuita	Sí	
SansGUI Modeling and Simulation Environment	No -	Sí	Sí	Ambiente de ejecución SansGUI Run-Time	Sí	
SAS Software	No -	Sí	Sí	Los modelos se pueden almacenar y compartir como conjuntos de datos SAS	No	
ServiceModel	Sí Optimización a través del software OptQuest y/o SimRunner	Sí	Sí	Los modelos pueden verse con la versión ProModel Player, que es gratuita	No	
ShowFlow	Sí El modo experimental permite determinar el mejor de un conjunto de opciones	Sí	Sí	Todos los modelos se pueden ejecutar con la versión evaluación, que es gratuita	No	
SIGMA	Sí Implementación de una metodología para selección y optimización	Sí	Sí	Crea código fuente C de las simulaciones	Sí	
SimCAD Pro	Sí -	Sí	Sí	-	No	
SIMPROCESS	Sí SIMPROCESS incluye OptQuest, una herramienta de optimización de simulaciones	Sí	No	-	-	

Desarrollo de modelos (continuación)						
Software		Optimización (Especificar)	Reusabilidad de código	Ejecución de modelos (¿Puede verse un modelo completo sin el software para desarrollo?)	Herramientas para ejecutar modelos	¿Estas herramientas tienen un costo?
SIMUL8 Professional	Sí	OptQuest incluido con el software	Sí	Sí	Las simulaciones se pueden ver y editar con el SIMUL8 Viewer	No
SIMUL8 Standard	Sí	OptQuest se puede agregar por \$495	Sí	No	-	-
SLIM	Sí	Optimiza redes mediante un conjunto de algoritmos configurables	Sí	No	-	-
Supply Chain Builder	Sí	Supply Chain Builder permite optimizar la red antes de ejecutar el modelo	Sí	Sí	SDI ofrece una versión de ejecución, que permite correr modelos y modificar datos del modelo	Sí
VisSim	Sí	Permite definir funciones objetivo para optimizar	Sí	Sí	Permite empaquetar el diagrama actual para enviarlo por mail mediante un comando	No
Visual Simulation Environment	No	-	Sí	No	-	No

Software	Animación				Precio (en dólares US)	
	Animación	Animación en tiempo real	Exportación de animaciones para presentaciones	Software de animación compatible	Estándar	Versión estudiantil
Analytica	No	-	No	No	Professional: \$995, Enterprise: \$1,595, Decision Engine: \$3,995	Gratuita
AnyLogic 5.0	Sí	Sí	Sí	-	\$3,950-\$12,250	\$395-\$1,400
Arena	Sí	Sí	No	Sí	\$495-\$25,000	Gratuita
AutoMod	Sí	Sí	Sí	Sí	Promedio: \$24,000	\$25
Crystal Ball Professional Edition	Sí	Sí	No	No	\$1,490	\$69 anuales
Crystal Ball Standard Edition	Sí	Sí	No	No	\$685	\$49 anuales
DecisionPro	No	No	No	No	Personal: \$495, Professional: \$795	\$99
DecisionScript	Sí	Sí	No	No	Cliente: \$995, Servidor: \$3,995	50% descuento
eM-Plant	Sí	Sí	Sí	Sí	Standard: \$19,000, Professional: \$31,000	No disponible
Enterprise Dynamics 5.0	Sí	Sí	Sí	Sí	A partir de \$3,500	Gratuita
ExpertFit	-	-	-	-	\$495-\$995	\$325-\$795
Extend Industry	Sí	Sí	No	Sí	\$2,295	Académica: \$1,150, Estudiante: \$125
Extend OR	Sí	Sí	No	Sí	\$1,595	Académica: \$795, Estudiante: \$125
Extend Suite	Sí	Sí	Sí	Sí	\$3,995	Académica: \$1995, Estudiante: \$125
Factory Explorer	No	No	No	Sí	\$29,800	\$295
FirstSTEP Designer	Sí	No	No	No	\$2,950	\$1,500
Flexsim	Sí	Sí	Sí	Sí	\$19,500	Académica: \$250 - \$500, Estudiante: Gratuita
GAUSS	No	-	-	-	No disponible	No disponible
GoldSim	No	Sí	No	-	\$3,950	Académica: Gratuita, Investigación: \$950
GPSS World for Windows	No	-	-	-	\$4,500	Gratuita
HighMAST™	Sí	Sí	No	Sí	No disponible	No disponible
MAST	Sí	Sí	No	No	\$7,500	\$750
MedModel	Sí	Sí	No	-	\$18,500	Académica: \$995, Estudiante: \$30
Micro Saint	Sí	Sí	No	No	\$8,995	Académica: \$1,500; Estudiante: \$60
mystrategy	No	Sí	No	-	£65.00	No disponible
NAG C Library	No	No	No	-	\$995+	\$395+
NAG SMP Library	No	No	No	-	\$995+	\$395+
PASION Simulation System	Sí	Sí	No	No	\$90	\$20
PIMSS	Sí	Sí	Sí	-	No disponible	No disponible
ProcessModel	Sí	Sí	No	Sí	\$2,395	Académica: \$995, Estudiante: \$50
ProModel	Sí	Sí	No	-	\$18,500	Académica: \$995, Estudiante: \$30

Software	Animación				Precio (en dólares US)	
	Animación	Animación en tiempo real	Exportación de animaciones para presentaciones	Software de animación compatible	Estándar	Versión estudiantil
Proplanner Manufacturing Process Management Software	No	No	No	No	Internet: \$500/mes, Cliente/Servidor: \$25,000+	Cliente/Servidor: \$500
ProVision	Sí	Sí	No	No	No disponible	No disponible
Resource Manager	No	No	-	-	\$1,200	\$125
SAIL	Sí	Sí	No	-	\$15,000	\$1,500
SansGUI Modeling and Simulation Environment	Sí	Sí	No	No	\$599	Académica: \$299, Estudiante: \$99
SAS Software	Sí	Sí	No	-	Varía	Varía
ServiceModel	Sí	Sí	No	-	\$18,500	Académica: \$995, Estudiante: \$30
ShowFlow	Sí	Sí	No	Sí	£345/\$570	Académica: £500, Estudiante: £250
SIGMA	Sí	Sí	Sí	Sí	\$1,850-\$5,000	Sólo con cursos, \$35/estudiante
SimCAD Pro	Sí	Sí	No	-	\$995	-
SIMPROCESS	Sí	Sí	Sí	-	\$9,995	Clase de 20 usuarios: \$750
SIMUL8 Professional	Sí	Sí	No	No	\$3,995	\$1,450
SIMUL8 Standard	Sí	Sí	No	No	\$1,295	Académica: \$1,450, Estudiante: Gratuita
SLIM	Sí	Sí	Sí	-	No disponible	No disponible
Supply Chain Builder	Sí	Sí	No	No	No disponible	No disponible
VisSim	Sí	Sí	Sí	-	Professional: \$2,500, Personal: \$495	Professional: \$375, Personal: \$75
Visual Simulation Environment	Sí	Sí	No	No	\$1,999	\$999

Tabla 8.1: Lista y características de lenguajes de simulación¹

Dos lenguajes muy importantes no fueron incluidos en esta tabla. Son ithink y Powersim:

- ithink es un paquete de software que provee una interfase gráfica para el desarrollo de modelos de dinámica de sistemas. Soporta directamente los diagramas de stock y flow, que se utilizan en la bibliografía de dinámica de sistemas. ithink se aplica en organizaciones para describir procesos, mercados, clientes y competidores; definir una estrategia operativa; e implementar cambios en forma efectiva.
- Powersim es un paquete de software que permite crear modelos de dinámica de sistemas y simuladores de negocios. Se puede utilizar para crear simuladores de negocios para experimentar con estrategias corporativas; análisis de operaciones, de marketing y competitivo; y toma de decisiones. Utilizando un modelo desarrollado en Powersim se pueden ver los efectos de distintas decisiones, descubrir potenciales problemas, y hacer ajustes. Las principales áreas de aplicación de Powersim son las áreas de industria y negocios, para tareas de planeamiento estratégico, administración de recursos, planeamiento y manejo de crisis y reingeniería de procesos.

9. Anexo C: Extend: Características Avanzadas

Extend posee gran cantidad de características y herramientas avanzadas, que permiten el desarrollo de modelos de simulación complejos. A continuación se describirán las más importantes, según la versión a partir de la cual están disponibles.

✍ 9.1. Extend CP

Extend CP es la versión básica de Extend, que permite el modelado de sistemas continuos. Las principales características y herramientas avanzadas que ofrece son:

9.1.1. Clonación de ítems

La clonación de ítems permite acceder en forma rápida a distintos parámetros de distintos cuadros de diálogo. En secciones anteriores se describió como clonar ítems al Notebook del modelo, ahora se verá que también se pueden clonar al modelo mismo. Esto permite acceder a distintos parámetros sin tener la necesidad de abrir los cuadros de diálogo correspondientes.

El mismo ítem de un cuadro de diálogo puede ser clonado a más de un destino, con lo cual se puede acceder de forma más rápida y cómoda a los parámetros más utilizados, poniéndolos tanto en el modelo como en el Notebook del modelo.

La clonación de ítems también facilita la modificación de valores durante la ejecución, ya que se evita tener que abrir cuadros de diálogo para cambiar parámetros.

En la Figura 9.1 se puede ver el ítem Delay del cuadro de diálogo 3, que fue clonado tanto dentro del modelo como en el Notebook del modelo. Cualquier modificación a cualquiera de las instancias de este valor modificará a las otras automáticamente.

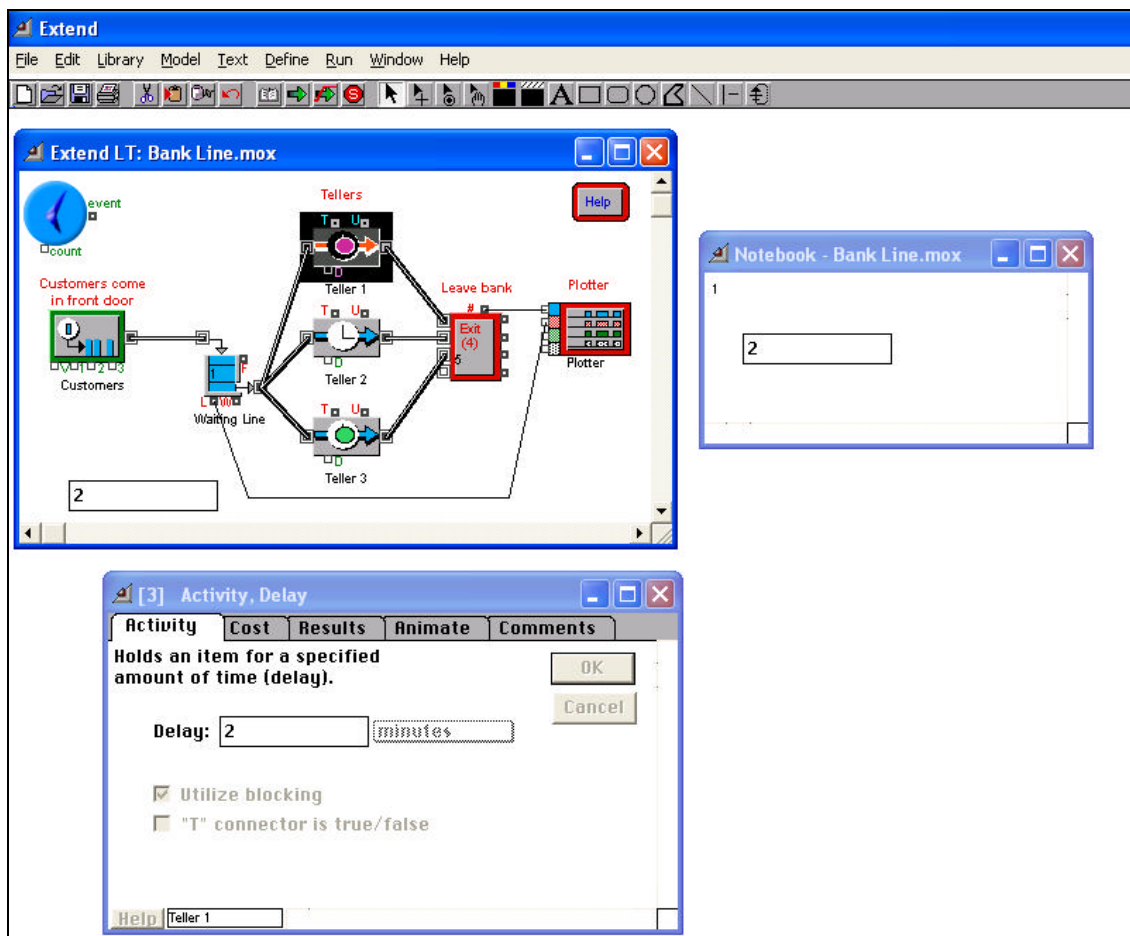


Figura 9.1: Ejemplo de clonación de ítems

9.1.2. Animación

Cuando se observa la salida de un Plotter luego de ejecutar un modelo, se puede ver como los valores cambian a través del tiempo. Si se deja el Plotter abierto durante la ejecución, se puede ver el progreso de los valores en el tiempo. Sin embargo, puede ser necesario tener información adicional mientras se ejecuta el modelo. Para esto, se pueden utilizar las capacidades de animación de Extend.

Se pueden realizar animaciones mediante la animación integrada, una animación personalizada, o con un paquete de animación como Proof Animation de Wolverine Software.

1. Animación integrada

Muchos de los bloques de Extend contienen animación integrada. La ayuda de estos bloques informa si tienen animación y, si la tienen, qué aspectos están animados:

- Animación en el icono del bloque: Consiste en animaciones dentro del bloque mismo, como por ejemplo un *Holding Tank*, que muestra como sube y baja el nivel de contenido.

- Animación entre iconos de bloques: Las librerías para modelos de eventos discretos son capaces de representar el flujo de ítems entre los bloques mediante imágenes o iconos que viajan a través de las conexiones.

2. Animación personalizada

Se puede agregar una animación personalizada a un modelo utilizando los bloques de la librería *Animation* o usando las funciones de animación en bloques propios que fueron creados:

- Bloques de la librería *Animation*: Los bloques *Animate Value*, *Animate Item* y *Animate Attribute* muestran animaciones personalizadas en respuesta a condiciones del modelo. Basándose en las opciones ingresadas a sus cuadros de diálogo, estos bloques pueden mostrar el valor que se ingresa, mostrar un nivel que cambia entre un mínimo y un máximo especificado, mostrar texto o animar una imagen cuando el valor ingresado excede un valor crítico, o cambiar la animación de un ítem según el valor de un atributo.
- Funciones de animación: Si se construyen bloques nuevos, Extend ofrece un conjunto de funciones que permiten diseñar la forma en que un bloque va a ser animado.

9.1.3. Análisis de sensibilidad

Un análisis de sensibilidad permite realizar experimentos e investigar los efectos de cambios en forma estructurada y controlada. Esto se hace ejecutando una simulación muchas veces y cambiando el valor de una variable o parámetro cada vez que la simulación es ejecutada. Cuando se ejecuta una simulación que tiene parámetros que cambian en cada ejecución, se puede ver el rango de los resultados y analizar tendencias o anomalías.

El análisis de sensibilidad provisto por Extend permite especificar qué parámetros cambiar y provee varios métodos para cambiarlos. Por ejemplo, se puede analizar el efecto que tiene en la cola de un banco la modificación del tiempo que tarda un cajero en atender a un cliente.

Para realizar un análisis de sensibilidad en un modelo, los pasos son los siguientes:

1. Elegir un parámetro a analizar.
2. Elegir cómo va a cambiar ese parámetro.
3. Seleccionar el número de ejecuciones para el análisis.
4. Opcionalmente, agregar un *Plotter MultiSim* o *Plotter DE MultiSim* para ver los resultados simultáneamente.
5. Ejecutar la simulación con la opción *Use Sensitivity Analysis* tildada.

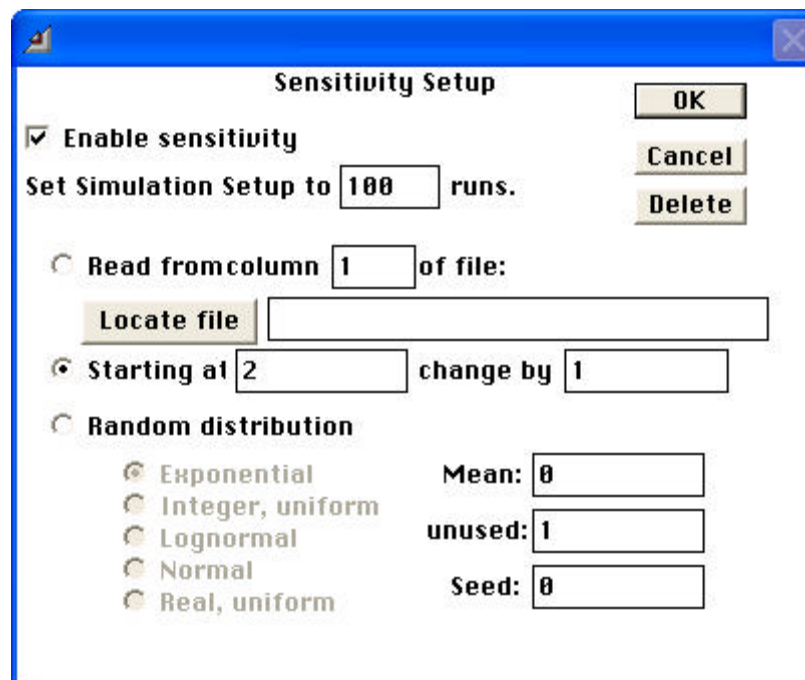


Figura 9.2: Cuadro de diálogo para análisis de sensibilidad

9.1.4. Optimización

La optimización es una herramienta poderosa, que puede determinar valores ideales para parámetros del modelo en forma automática. La forma de realizar esto es mediante la ejecución del modelo varias veces, modificando parámetros, hasta encontrar una solución aceptable que cumpla con las condiciones especificadas. El optimizador provisto por Extend utiliza un algoritmo evolutivo que reduce el número de ejecuciones necesarias para encontrar una solución satisfactoria.

La mayoría de los modelos tienen parámetros que especifica el modelador y que no varían, como por ejemplo el tiempo de una actividad. Algunos parámetros, por ejemplo el número de máquinas que realizan una actividad, pueden variar y cambiar la eficiencia del modelo. Para encontrar la solución con el mejor costo-beneficio, se podría modificar manualmente el modelo y cambiar el número de máquinas para cada ejecución hasta encontrar una solución satisfactoria. Esto podría llevar mucho tiempo, y puede evitarse utilizando la optimización en forma automática.

Por ejemplo, utilizar más máquinas puede disminuir el trabajo sin finalizar, pero esas máquinas tienen un costo, y quizás el costo de las mismas sea mayor que el beneficio generado por el trabajo extra finalizado. Sabiendo estos costos, se puede crear una simple ecuación de costos y ejecutar el modelo varias veces, buscando que el costo sea minimizado. La optimización automatiza esta tarea, evitando que el modelador deba ejecutar el modelo infinidad de veces. Además, las ecuaciones de costos pueden tener restricciones, que permiten limitar el valor de alguna variable según el valor de otras, y esto puede manejarse mediante la optimización.

Extend provee el procedimiento de optimización en un bloque que simplemente se agrega al modelo. Este bloque controla todos los aspectos de la optimización. El hecho de que la optimización se implemente en un bloque provee gran flexibilidad, ya que el modelador puede acceder al código del bloque y modificarlo, creando su propio bloque optimizador personalizado.

La optimización es una técnica que permite encontrar la mejor solución a un problema en forma automática. El problema se define como una función objetivo, que equivale a una ecuación de costo o beneficio, que el modelador está tratando de minimizar o maximizar evitando el tedio de intentar distintos valores en cada ejecución manualmente.

La mayoría de los algoritmos de optimización que pueden resolver modelos estocásticos utilizan una población inicial de posibles soluciones. Cada posible solución se ejecuta en el modelo varias veces, calculando promedios, y ordenando las soluciones. La mejor solución de la población es elegida, y modificada ligeramente para crear un nuevo conjunto de soluciones. Cada conjunto de soluciones se denomina generación. Este proceso se repite hasta que probablemente no se vayan a encontrar mejores soluciones. Entonces se detiene, y se informa de la mejor solución hallada.

Los pasos para optimizar un modelo en Extend son:

- Abrir el modelo que se desea optimizar.
- Abrir la librería *Generic*.
- Agregar un bloque *Evolutionary Optimizer* al modelo.
- Definir la función objetivo a optimizar.
- Determinar los parámetros necesarios para la función objetivo.
- Arrastrar las variables necesarias al bloque *Evolutionary Optimizer*.
- Determinar los límites para esas variables en la tabla *Optimizer's Variables* del bloque.
- Escribir la función objetivo en el cuadro de diálogo del bloque.
- Agregar las restricciones que correspondan.
- Configurar el optimizador para modelo aleatorio o no aleatorio.
- Ejecutar el modelo.

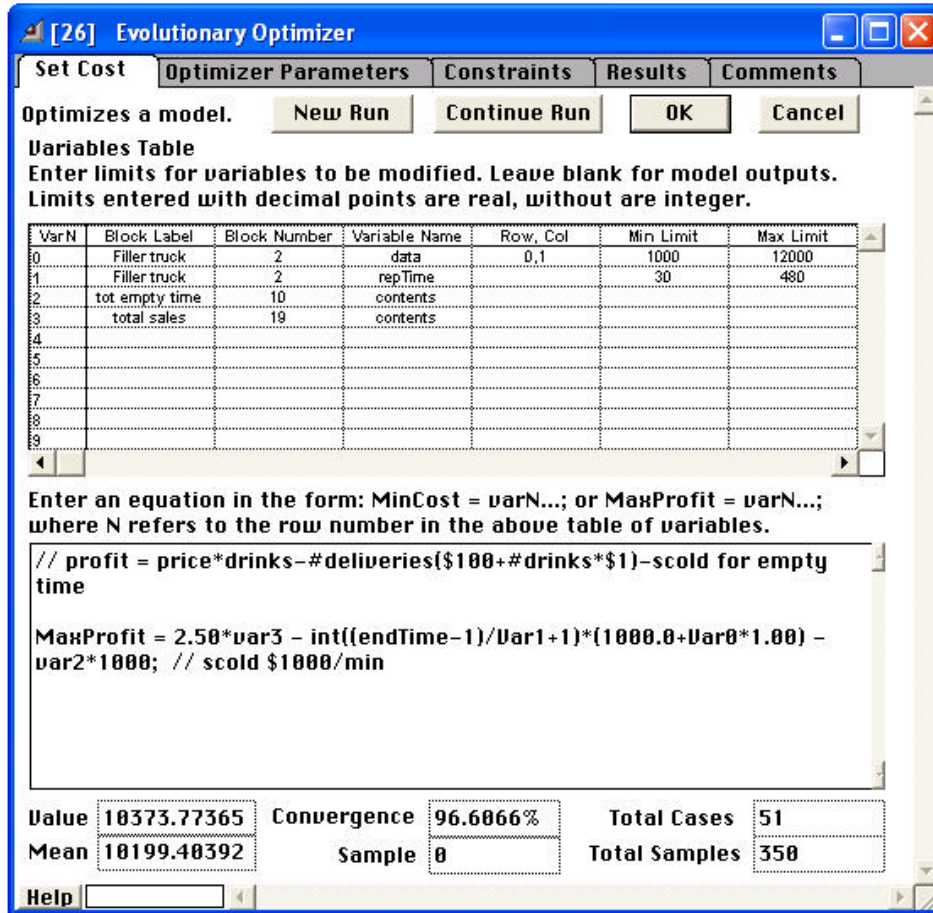


Figura 9.3: Cuadro de diálogo para optimización

9.1.5. Edición de ecuaciones

Las librerías de bloques de Extend permiten construir modelos rápidamente. Sin embargo, puede que estos bloques no satisfagan las necesidades del modelador. Por ejemplo, puede no existir un bloque que provea la función o ecuación que se necesita. O puede ser necesario combinar las funciones de varios bloques en uno. Las posibles soluciones a estos problemas son:

- Seleccionar varios bloques y convertirlos en un bloque jerárquico, opción que ya fue descrita.
- Modificar la estructura del bloque, accediendo a su código y modificándolo. Esta opción será descrita en la sección 4.7 Lenguaje de Programación.
- Usar el bloque *Equation* de la librería *Generic* o *Equation DE* de la librería *Discrete Event* para combinar funciones directamente, según el modelo sea continuo o discreto.

Los bloques *Equation* y *Equation DE* permiten ingresar una ecuación directamente al cuadro de diálogo de un bloque. La ecuación debe ser de la forma *salida = fórmula*;

La ecuación es compilada automáticamente cuando se hace click en OK.

Los bloques *Equation* y *Equation DE* son similares en varios aspectos a la barra de fórmulas de una hoja de cálculo. La mayoría de los componentes usuales son los mismos (operadores, valores, funciones, etc.). La diferencia más importante es que en vez de hacer referencia a una celda, estos bloques tienen conectores de entrada y salida identificados por nombre.

Ambos bloques de ecuaciones tienen cinco conectores de valor de entrada y uno de salida. El bloque *Equation DE* también tiene un conector de ítem de entrada y uno de salida. Este bloque tiene la habilidad de leer atributos de los ítems que pasan a través de él, y usar esos valores en la ecuación. Dentro del cuadro de diálogo de ambos bloques de ecuaciones, se puede ingresar una ecuación que use cualquiera de las funciones y operadores de Extend sobre los valores de entrada. La ecuación puede ser tan simple como una operación matemática, o tan compleja como un segmento de programa completo.

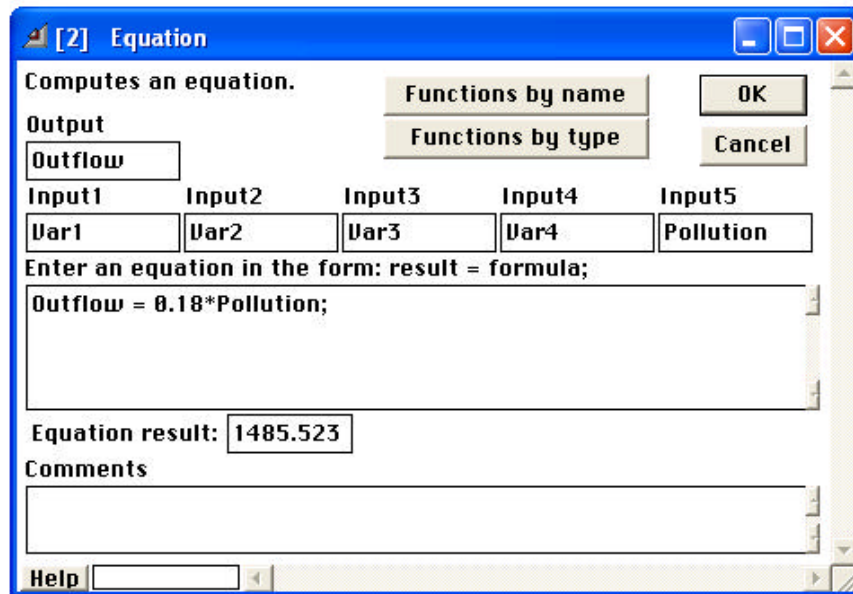


Figura 9.4: Cuadro de diálogo para ecuaciones

9.1.6. Bloques de control

Extend posee tres bloques especiales a los que se puede acceder desde el menú *Model, Controls*. Estos bloques se utilizan para agregar control interactivo al modelo. Se usan para controlar otros bloques y mostrar valores directamente mientras se ejecuta el modelo.

Estos controles son *Slider*, *Switch* y *Meter*. Los bloques *Slider* y *Switch* se usan para determinar valores en los modelos. El bloque *Meter* para mostrar valores mientras el modelo se está ejecutando.

El bloque *Slider* consiste en un deslizador, que puede ser movido para cambiar la salida. Se determinan los valores máximo y mínimo ingresándolos en el icono del bloque. Se puede cambiar el valor de salida del *Slider* moviendo el indicador de nivel hacia arriba o hacia abajo. Este bloque tiene tres conectores de salida, uno para el valor actual, otro para el máximo y otro para el mínimo. El bloque *Slider* es útil cuando, durante la ejecución de la simulación, se desea generar un valor de salida en un rango determinado, y éste puede no ser exacto.

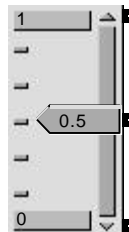


Figura 9.5: Bloque *Slider*

El bloque *Switch* tiene dos entradas y una salida, y tiene el aspecto de un botón de encendido común. La salida del *Switch* es uno o cero, según esté encendido o apagado. Haciendo click sobre la derecha del botón se enciende, y sobre la izquierda se apaga. Este bloque es útil para controlar, durante la ejecución de la simulación, un bloque que tiene entrada de tipo verdadero-falso, ya que Extend toma cero como falso y uno como verdadero. Los conectores de entrada sirven para modificar el estado del *Switch* desde adentro del modelo. Se encuentran uno a la derecha y otro a la izquierda del *Switch*, y cuando uno de ellos recibe un valor verdadero (mayor o igual a 0.5) se activa su lado del *Switch*.



Figura 9.6: Bloque *Switch*

El bloque *Meter* muestra valores que varían entre un mínimo y un máximo. Los valores mínimo y máximo se modifican en el cuadro de diálogo del bloque, o mediante la conexión de bloques a los conectores de mínimo y máximo. El conector de valor actual es el que se utiliza para recibir el valor que se muestra.

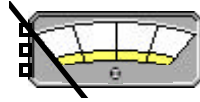


Figura 9.7: Bloque *Meter*

9.2. Extend OR

Extend OR es la versión de simulación de eventos discretos de Extend. Extend OR incluye todas las funcionalidades de Extend CP. Las principales características y herramientas avanzadas que agrega están contenidas en sus librerías, que permiten el modelado de sistemas de eventos discretos. Estas librerías son:

9.2.1. Discrete Event

La librería *Discrete Event* provee los bloques básicos para modelar un sistema de eventos discretos.

Sus bloques se dividen en diez grupos: Ejecutivo, Actividades, Atributos, Trabajo en lotes, Generadores, Información, Colas, Recursos, Encaminamiento y Estadísticas.

- Ejecutivo: Es el bloque de control de toda simulación de eventos discretos. Siempre debe estar presente en este tipo de simulación, ya que en él se especifica la duración de la simulación.
- Actividades: Son tres bloques que detienen ítems por un período de tiempo especificado, y luego los liberan.
- Atributos: Son bloques que permiten darle identidad a los ítems, cambiando sus características y operando de distinta manera según las características de los mismos.
- Trabajo en lotes: Son dos bloques que permiten unir y dividir ítems.
- Generadores: Son dos bloques que proveen ítems con tiempos entre arribos especificados.
- Información: Son bloques que muestran información acerca de los ítems que pasan por ellos y acerca de otros bloques.
- Colas: Son bloques para mantener, ordenar y liberar ítems.
- Recursos: Son bloques que proveen ítems genéricos.
- Encaminamiento: Son bloques que permiten mover ítems al lugar deseado.
- Estadísticas: Son bloques que reportan datos estadísticos acerca de los ítems que pasan por ellos y acerca de otros bloques.

9.2.2. Manufacturing

La librería *Manufacturing* contiene un conjunto de bloques para modelar sistemas de eventos discretos industriales y comerciales. Sus bloques permiten realizar tareas más específicas que las de la librería *Discrete Event*.

Sus bloques se dividen en seis grupos: Actividades, Trabajo en lotes, Generadores, Colas, Recursos y Encaminamiento.

- Actividades: Son eventos que le ocurren a recursos, como por ejemplo operaciones en una máquina, esperas a clientes o movimiento a través de cintas transportadoras. Estos bloques regulan el flujo de ítems a través del modelo. Cada bloque pide un ítem, lo procesa, y lo mantiene hasta que el siguiente bloque está listo para recibirlo.
- Trabajo en lotes: Son bloques que representan situaciones donde los ítems son ensamblados o unidos para formar un solo ítem, ya sea en forma temporaria o permanente. También hay bloques que permiten desensamblar una unidad en los ítems que la componen.
- Generadores: Proveen ítems. Son dos, *Schedule* y *DownTime*. *Schedule* provee ítems de acuerdo a una planificación. *DownTime* provee ítems a intervalos aleatorios y se utiliza para simular roturas y mantenimientos no programados.
- Colas: Mantienen ítems de recursos y los liberan en el orden especificado para luego ser utilizados por operaciones de manufactura.
- Recursos: Son proveedores de ítems específicos para la manufactura, como por ejemplo herramientas, clientes, o trabajadores. Los Recursos pueden ser ilimitados, proveyendo ítems sin parar mientras dure la simulación. También pueden ser limitados, especificando el número de ítems disponibles.
- Encaminamiento: Es un conjunto de bloques que permiten controlar el flujo de ítems con mayor detalle que la librería *Discrete Event*.

9.2.3. QuickBlocks

La librería *QuickBlocks* contiene un conjunto de bloques jerárquicos que proveen soluciones a problemas comunes. Se utilizan como cualquier otro bloque, conectándolos a los bloques necesarios y modificando sus parámetros.

Algunas de las operaciones que realizan son vaciar una cola a intervalos periódicos, seleccionar la cola más corta, leer atributos de archivos, etc.

9.3. Extend Industry

Extend Industry es la versión de Extend para proyectos de gran envergadura. Extend Industry incluye todas las funcionalidades de Extend CP y Extend OR. Las principales características y herramientas avanzadas que agrega son una base de datos y la arquitectura de flujo.

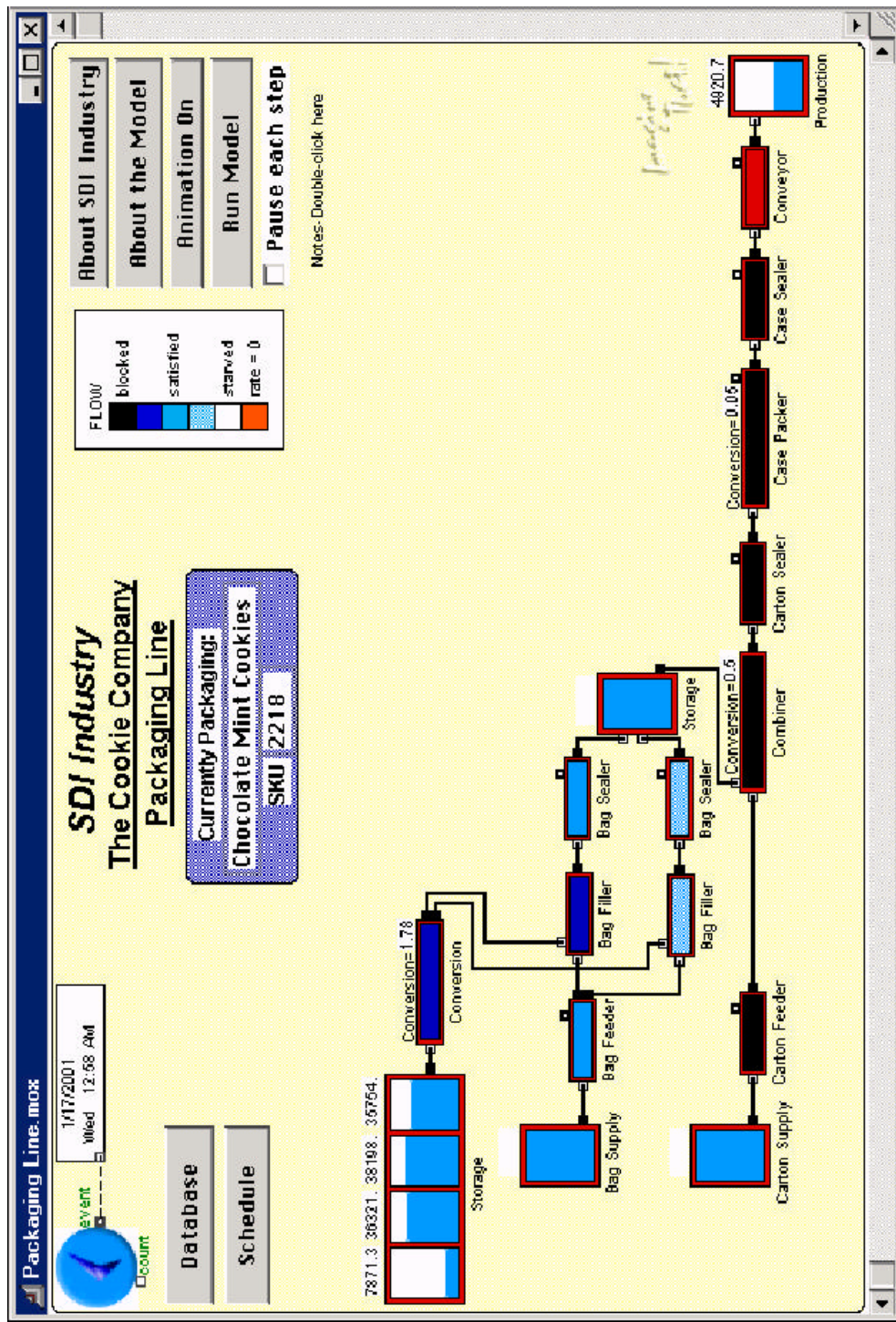


Figura 9.8: Ejemplo de modelo en Extend Industry

9.3.1. Base de datos integrada

Extend Industry incluye una base de datos integrada, que permite crear, ver y administrar todos los datos del modelo, configurar las tablas para reportes y experimentos, usar atributos relacionados con la base de datos, manejar formatos como fecha, hora, y moneda, y utilizar bloques relacionados con la base de datos.

Los beneficios provistos por el uso de una base de datos integrada son la flexibilidad y facilidad de uso que provee la manipulación centralizada de los datos del modelo.

Industry incluye bloques que permiten interactuar con la base de datos en las librerías *SDI Tools*, *Flow e Items*. El sistema de administración de base de datos utiliza wizards y editores de tablas que facilitan la creación y administración de la base de datos desde adentro de Extend.

Para quienes prefieren administrar los datos desde Excel, Extend Industry provee un agregado para Excel que permite transferir datos desde y hacia el modelo en Extend. Este agregado permite la creación de DB Workbooks que se pueden utilizar como front-end.

Extend también soporta la importación y exportación de datos con Access, mediante un bloque especial que se agrega al modelo.

9.3.2. Arquitectura de flujo

La arquitectura de flujo provee un conjunto de bloques que permiten modelar procesos donde el flujo es tratado como un ratio, como por ejemplo llamadas por hora, botellas por minuto, o transacciones por segundo. Estos bloques se encuentran en una librería llamada *Flow*, que es utilizada generalmente para modelar líneas de empaque de alta velocidad, procesamiento continuo de alto volumen, o cualquier tipo de proceso donde el flujo es tratado como un ratio.

La librería *Discrete Event* de Extend representa los materiales como unidades discretas, objetos o ítems. Los bloques de *Industry Flow* representan los materiales como ratios de flujos. Usando Industry, se puede modelar fácilmente sistemas de flujo de materiales combinando bloques de *Discrete Event* con bloques de *Flow*.

La arquitectura de flujo es muy diferente de la arquitectura clásica de eventos discretos. Sin embargo, *Flow* es totalmente compatible con los bloques de eventos discretos de Extend. De hecho, la librería *Flow* incluye bloques que permiten realizar conversiones de flujo a ítems y viceversa.

La arquitectura de flujo es útil cuando se necesita un modelo orientado a la logística, donde el flujo de material es continuo, o donde es más fácil considerar al flujo como continuo. En estas situaciones, los sistemas clásicos continuos o de eventos discretos no serían apropiados. La arquitectura de flujo permite construir modelos que se representan como redes de flujo continuo de materiales.

9.4. Extend Suite

Extend Suite es un conjunto de herramientas integradas completo. Un software de análisis de última generación con el paquete Extend Industry. Extend Suite se utiliza cuando se requiere administración de datos intensiva, ajustes a distribuciones de probabilidad mediante Stat::Fit, o animaciones a nivel profesional mediante Proof Animation.

9.4.1. Stat::Fit

Existen casos donde no se sabe qué distribución de probabilidad es la más adecuada para un determinado modelo, pero se tienen un conjunto de datos empíricos que se desea modelar usando distribuciones aleatorias. Stat::Fit es un paquete de ajuste de distribuciones de probabilidad. Este paquete sirve para determinar que distribución estadística se acerca más a los datos del mundo real disponibles.

En los modelos de simulación, suele ser útil caracterizar una entrada aleatoria con una distribución de probabilidad (por ejemplo, el tiempo entre arribos). Generalmente esto requiere datos históricos que documenten el comportamiento del sistema y un análisis para determinar la distribución de probabilidad que mejor se adapta a los datos. Usar distribuciones estadísticas en vez de datos históricos para las entradas provee dos ventajas:

- Los valores de las entradas no están limitados a lo que sucedió históricamente.
- Para distribuciones continuas, existe un conjunto infinito de posibles entradas.

Stat::Fit de Geer Mountain Software determina qué distribución de probabilidad, si hay alguna, se ajusta mejor a los datos del mundo real provistos. La función Auto::Fit automáticamente ajusta los datos a distribuciones continuas o discretas, y provee comparaciones relativas entre las distribuciones. Las distribuciones resultantes pueden ser exportadas directamente a Extend. Existe una interfase que le facilita al modelador el acceso a Stat::Fit desde Extend.

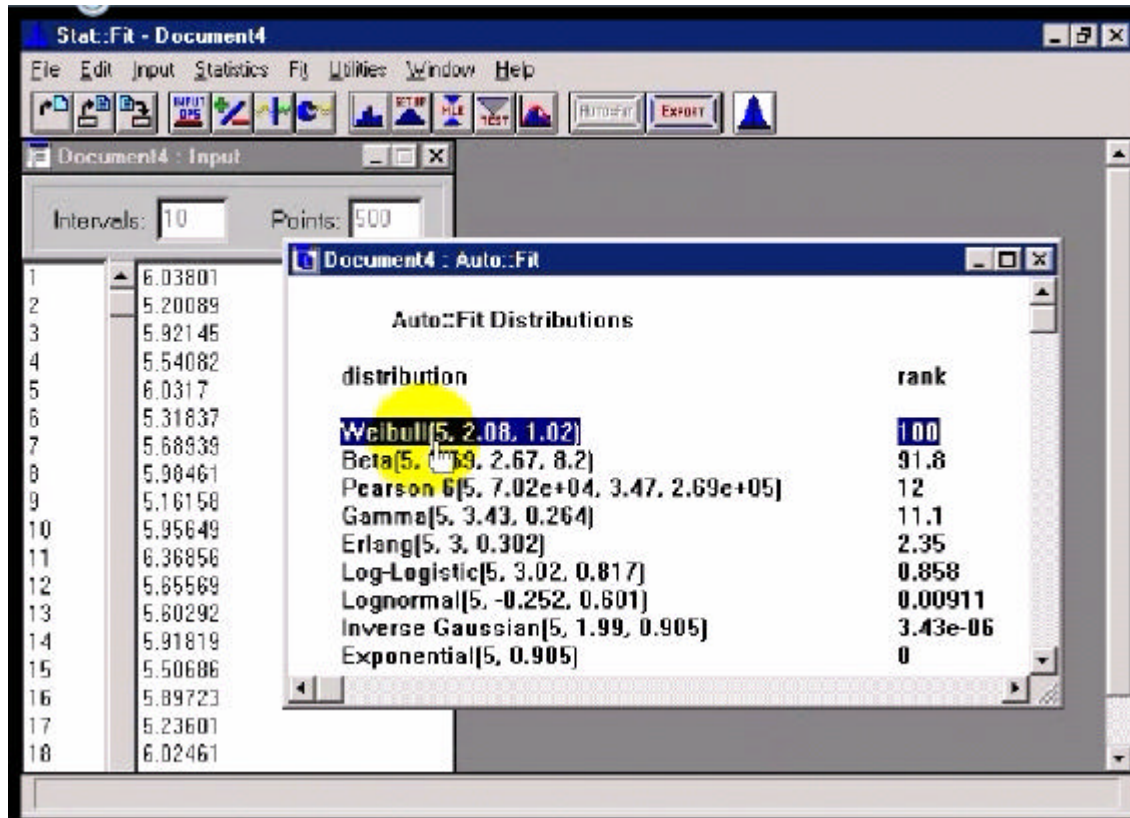


Figura 9.9: Selección de una distribución a partir de datos empíricos en Stat::Fit

9.4.2. Proof Animation

Proof Animation de Wolverine Software permite crear animaciones avanzadas para presentaciones a los usuarios. Permite crear animaciones vectoriales, que muestran distribuciones físicas y recorridos. Se puede controlar el paso de la animación e investigar comportamientos inusuales mediante aceleraciones, retrasos, o saltos. Se pueden realizar zooms en cualquier momento a cualquier nivel. Permite importar objetos CAD de archivos dxf. Proof Animation usa DirectX para mostrar animaciones de gran número de objetos y provee las características necesarias para realizar animaciones.

Las principales características y beneficios de Proof Animation son:

Característica	Beneficio
Los objetos se mueven en forma suave.	Las actividades que incluyen movimiento son mostradas en forma realista.
Ratio constante entre el tiempo de la animación y el tiempo del modelo.	La animación no se acelera o retarda en forma artificial.
Movimiento con dirección y objetos que pueden rotar.	Los objetos rotan, en vez de deslizarse, al llegar a esquinas.
Animación automática de colas.	Los ítems se apilan al esperar en colas.
Escalas físicas precisas.	La geometría del modelo es representada en forma exacta.
Múltiples vistas de la animación.	Se pueden crear vistas para focalizarse en el área de interés.
Animación con pantalla dividida.	Se pueden ver varias vistas de la animación a la vez.
Vistas isométricas.	Da un aspecto tridimensional a la animación.

Tabla 9.1: Características y beneficios de Proof Animation

Proof Animation puede trabajar con programas de simulación continuos o de eventos discretos. Imagine That, Inc. agregó una interfase a Proof Animation que hace que la transferencia de información del modelo sea mucho más fácil que con otras herramientas de simulación.

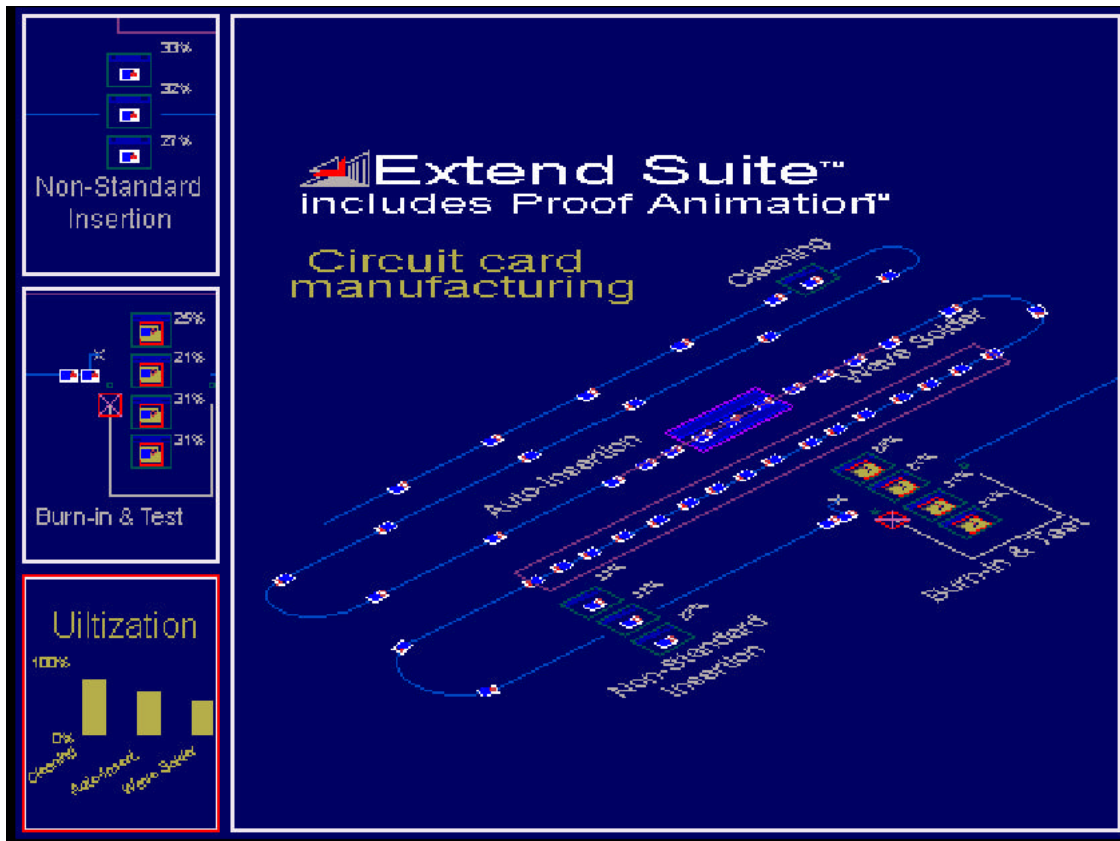


Figura 9.10: Ejemplo de animación de Proof Animation

10. Anexo D: El Lenguaje ModL

A continuación se describe en forma más detallada el lenguaje ModL. Se analiza la estructura, los manejadores de mensajes, las variables, vectores y constantes, y las funciones.

10.1. Estructura

✂ 10.1.1. Nombres

Los nombres de variables, constantes, funciones y procedimientos pueden tener hasta 63 caracteres. Pueden tener letras, números y guiones bajos (_). Deben comenzar con una letra o guión bajo. Algunos nombres están reservados por el sistema. ModL es un lenguaje insensible a mayúsculas y minúsculas, por lo tanto myName es equivalente a myname y a MYNAME.

✂ 10.1.2. Tipos de datos

ModL provee cinco tipos de datos:

- real o double: Para representar números reales. Equivalente a double de C.
- int o long: Para representar números enteros. Equivalente a long de C.
- str15: Cadena de 15 caracteres.
- str31: Cadena de 31 caracteres.
- string o str255: Cadena de 255 caracteres.

Para los tipos de datos que tienen más de un nombre, estos son intercambiables (por ejemplo, int y long se pueden usar indistintamente).

✎ **10.1.3. Conversiones de tipos**

Generalmente, ModL realiza todas las conversiones de tipo necesarias automáticamente. Por lo tanto, se pueden asignar valores enteros a variables reales sin realizar conversiones explícitas. Usar valores enteros para expresiones de números reales es peligroso y puede dar resultados erróneos. Por ejemplo,

```
z = 1 / 2 * a;
```

dará como resultado siempre cero, ya que la división del entero 1 por el entero 2 da como resultado 0. La forma correcta de escribir esta expresión sería

```
z = 1.0 / 2.0 * a;
```

donde 1.0 y 2.0 serán interpretados como valores reales, y su división dará como resultado 0.5.

✎ **10.1.4. Vectores**

Los vectores en ModL pueden tener hasta cinco dimensiones, y pueden ser de cualquier tipo de datos. El límite por índice es de dos mil millones de elementos. ModL provee tanto vectores de tamaño fijo como variable. Por ejemplo,

```
real myArray [ ] [4];
```

declara un vector bidimensional de número reales, con número variable de filas y cuatro columnas.

Existen funciones para consultar y modificar (en el caso de vectores de tamaño variable) el tamaño de un vector.

ModL también provee un conjunto de funciones que permite trabajar con listas encadenadas.

✎ **10.1.5. Operadores**

A continuación se describe el conjunto de operadores provisto por ModL.

Operadores de asignación	
Operador	Descripción
id = expresión;	Asignación
id += expresión;	Equivalente a id = id + expresión
id -= expresión;	Equivalente a id = id - expresión
id *= expresión;	Equivalente a id = id * expresión
id /= expresión;	Equivalente a id = id / expresión
id++;	Equivalente a id = id + 1
++id;	También equivalente a id = id + 1
id--;	Equivalente a id = id - 1
--id;	También equivalente a id = id - 1
Operadores matemáticos	
Operador	Descripción
+	Adición
-	Sustracción
*	Multiplicación
/	División
^	Exponencial
%	Módulo
MOD	Módulo
Operadores lógicos	
Operador	Descripción
AND o &&	Y lógico
OR o	O lógico
NOT o !	NOT lógico
!= o <>	Distinto
<	Menor
<=	Menor o igual
>	Mayor
>=	Mayor o igual
==	Igual

Tabla 10.1: Operadores

✍ 10.1.6. Expresiones de control

Las expresiones de control provistas por ModL son las siguientes:

Expresión	Uso
ABORT	ABORT;
BREAK	BREAK;
CONTINUE	CONTINUE;
DO – WHILE	DO Expresión; WHILE (comparación);
FOR	FOR (inicialización; comparación; incremento) Expresión;
GOTO	GOTO nombre; ... nombre: ...
IF	IF (comparación) Expresión;
IF-ELSE	IF (comparación) Expresión A; ELSE Expresión B;
RETURN	RETURN; o RETURN (valor);
SWITCH	SWITCH(expresión) { CASE valorentero1: Expresiones; BREAK; CASE valorentero2: Expresiones; BREAK; DEFAULT: Expresiones; BREAK; }
WHILE	WHILE (comparación) Expresión;

Tabla 10.2: Expresiones de control

Múltiples expresiones pueden encerrarse entre llaves:

```
{
expresión A;
expresión B;
}
```

✍ 10.1.7. Procedimientos y funciones definidas por el usuario

Los procedimientos y funciones definidas por el usuario le permiten al usuario crear su propio código reusable, y hacer que su código sea más comprensible.

La diferencia entre un procedimiento y una función es que un procedimiento no retorna ningún valor, en cambio una función sí. Las funciones y procedimientos deben ser definidos antes de ser utilizados.

ModL permite sobrecargar funciones y manejadores de mensajes. Esto permite que cuando se incluyen archivos que tienen funciones y manejadores de mensajes básicos que el programador necesite redefinir, éste pueda hacerlo.

- 10.2. Manejadores de mensajes

Los manejadores de mensajes permiten reaccionar ante la recepción de un mensaje. Existen dos tipos de mensajes, de sistema y de cuadros de diálogo.

Los mensajes de sistema son enviados por Extend cuando se construyen modelos, se ejecutan simulaciones o se utilizan funciones en ModL que explícitamente envían mensajes a otros bloques. Por ejemplo, cuando se agrega un bloque a un modelo, se envía el mensaje CreateBlock al nuevo bloque. Si existe un manejador para ese mensaje, es ejecutado; si no, no sucede nada. Algunos mensajes de sistema son enviados a todos los bloques, y otros sólo a un bloque específico. Por ejemplo, el mensaje InitSim, que indica que la simulación está comenzando, se envía a todos los bloques. En cambio, el mensaje CreateBlock sólo se envía al bloque se agregó al modelo.

Los mensajes de sistema son:

Mensaje	Momento de envío
AbortDialog-Message	Si el usuario detiene la ejecución de un manejador de mensajes o ese manejador de mensajes ejecuta un comando <i>Abort</i> , se envía este mensaje al bloque. Se utiliza como un manejador de excepciones.
AbortSim	Enviado después de una finalización prematura de la ejecución. Por ejemplo, por presionar el botón <i>Stop</i> o si ocurre un error. Un buen uso para AbortSim es resetear valores que pueden ser incorrectos debido a cálculos incompletos. EndSim siempre es ejecutado luego de AbortSim.
ActivateModel	Enviado a todos los bloques de un modelo cuando ese modelo es puesto delante de otro modelo. Esto permite que los bloques puedan modificar sus datos si el modelo es activado.
AnimationStatus	Enviado a todos los bloques de un modelo cuando el ítem <i>Show Animation</i> del menú <i>Run</i> cambia de estado.
AdviseReceive	Enviado al bloque cuando recibe datos actualizados a través de una conversación advise.
BlockClick	Enviado cuando el usuario hace click en el bloque.
BlockIdentify	Reservado para uso de Imagine That, Inc.
BlockLabel	Enviado al bloque al que se le acaba de cambiar el label.
BlockMove	Enviado a todos los bloques que se han movido cuando se termina de realizar el desplazamiento.
BlockRead	Este mensaje debe ser usado con cuidado, ya que puede producir problemas si se usa en forma incorrecta. Es usado para convertir tablas de datos de la versión 3.x a la versión 4.
BlockReceive0 a BlockReceive4	Mensajes personalizados. Son usados por la librería <i>Discrete Event</i> como mensajes de sistema. Se sugiere el uso de UserMgr0 a UserMgr9 para mensajes personalizados de usuario.
BlockReport	Enviado después de FinalCalc. Si un bloque recibe este mensaje, ha sido seleccionado para un reporte.
BlockTableInfo	Enviado por algunos bloques para consultar el tamaño de las tablas de datos.
BlockUndelete	Enviado a un bloque cuando su eliminación es revertida mediante el comando <i>Undo</i> .
CellAccept	Enviado cuando el usuario termina de editar una celda en alguna de las tablas de datos del bloque.
CheckData	Enviado después de que se ejecuta el comando de ejecución de simulación y antes de que la simulación comience. Este es el mejor lugar para validar si todos los datos que el bloque necesita son válidos. Si los datos son inválidos, se debería ejecutar un comando de cancelación de ejecución (<i>Abort</i>), para que Extend informe del problema.
ClearStatistics	Enviado cuando se deben resetear los valores estadísticos de un bloque.
CloseModel	Enviado cuando se cierra el modelo, antes de la decisión de guardar el modelo.
Connection-Make	Enviado a todos los bloques involucrados cuando el usuario crea una nueva conexión.
Connector-Name	Mensaje de conector. Enviado cuando el conector de un bloque recibe un mensaje de otro bloque usando las funciones de mensajes entre conectores.
Connector-ShowHide	Enviado cuando el usuario oculta o muestra los conectores.

ContinueSim	Este mensaje se envía en lugar de InitSim cuando se continúa una ejecución guardada del modelo.
CopyBlock	Enviado a todos los bloques seleccionados antes de una operación de copia.
CreateBlock	Enviado cuando el bloque es agregado al modelo. Este es el lugar adecuado para inicializar variables.
DataTable-Scrolled	Enviado al bloque correspondiente cuando se utiliza el scroll de una de sus tablas de datos.
DeleteBlock	Enviado cuando el usuario elimina un bloque del modelo.
DialogClick	Enviado cuando el usuario hace click en un ítem de diálogo.
DialogClose	Enviado cuando se cierra un cuadro de diálogo.
DialogItem-Refresh	Enviado cuando un ítem de diálogo se vuelve visible.
DialogOpen	Enviado cuando se abre el cuadro de diálogo de un bloque.
DragCloneTo-Block	Enviado cuando un ítem clonado es arrastrado y soltado sobre un bloque.
EndSim	Enviado al final de la simulación. Se utiliza para liberar memoria o resetear valores.
FinalCalc	Enviado después de que el último mensaje Simulate terminó y antes de los mensajes BlockReport y EndSim.
HBlockClose	Enviado cuando se cierra la estructura de un bloque jerárquico.
HBlockHelp-Button	Enviado a todos los bloques dentro de un bloque jerárquico cuando se presiona el botón de ayuda del bloque jerárquico.
HBlockMove	Enviado a todos los bloques dentro de un bloque jerárquico cuando el bloque jerárquico es movido.
HBlockOpen	Enviado a todos los bloques dentro de un bloque jerárquico cuando el bloque jerárquico es abierto.
HBlockUpdate	Enviado a todos los bloques dentro de un bloque jerárquico cuando el bloque jerárquico es modificado y cerrado.
HelpButton	Enviado cuando el usuario hace click en el botón de ayuda de un bloque.
InitSim	Enviado justo antes de que comience la simulación. Este es el momento apropiado para inicializar variables estáticas, ítems de diálogo, o conectores que cambian al iniciar la simulación.
ModelSave	Enviado a cada bloque al iniciarse una acción de guardar el modelo.
ModelOpen	Enviado cuando un modelo es abierto.
PasteBlock	Enviado cuando el usuario pega un bloque en el modelo.
PauseSimulation	Enviado a todos los bloques cuando el usuario pone la simulación en pausa.
Plotter0Close a Plotter4Close	Enviado cuando el usuario cierra la ventana de un plotter.
ProofAnimation	Enviado por los bloques que interactúa con Proof Animation para producir animaciones.
QueueFunction	Enviado por el bloque <i>QueueTools</i> y los bloques de cola para comunicarse entre ellos.
ResumeSim	Enviado cuando se continúa una simulación detenida. Sólo se envía a los bloques que fueron editados durante la pausa.
Simulate	Enviado a cada paso en la simulación. Este mensaje es enviado muchas veces, no sólo una. Aquí es donde la mayoría de la acción sucede.
StepSize	Enviado luego de todos los mensajes CheckData. Si el código ModL debería tener un StepSize particular, debería ser configurado aquí.
TabSwitch	Enviado cuando se cambia de lengüeta en un cuadro de diálogo.
UpdateStatistics	Enviado cuando un bloque necesita recalculer sus variables estadísticas.
UserMgr0 a UserMgr9	Mensajes personalizados definidos por el usuario. Enviado cuando un bloque recibe un mensaje de otro (conectado o no) mediante la función Send-MsgToBlock.

Tabla 10.3: Mensajes de sistema

Los mensajes de cuadros de diálogo son enviados al bloque cuando se hace click en controles del cuadro de diálogo, o se deseleccionan parámetros. Funcionan de manera similar a los mensajes de sistema. Por ejemplo, si un bloque tiene un botón Count, cuando se hace click en el mismo, Extend envía un mensaje Count al bloque. Si el bloque tiene un manejador para ese mensaje, éste se ejecuta; caso contrario no sucede nada.

Mensaje	Momento de envío
OK	Enviado cuando se hace click en el botón OK.
Cancel	Enviado cuando se hace click en el botón Cancel.
ButtonX	Si se creó un botón, botón de radio, o checkbox llamado ButtonX, este mensaje es enviado cuando se hace click en el mismo.
ItemY	En un cuadro de diálogo, cuando está seleccionado un parámetro o texto que puede editarse y se hace click en otro ítem o se presiona la tecla Tab (quitando la selección del ítem), se envía un mensaje con el nombre del ítem que estaba seleccionado. Esto es útil para verificar el valor de un ítem que pudo haber sido cambiado.

Tabla 10.4: Mensajes de cuadros de diálogo

- 10.3. Variables de sistema

Existen un conjunto de variables de sistema que dan información acerca del estado de la simulación. Estas variables pueden ser tanto consultadas como modificadas.

Nombre	Descripción
AnimationOn	Indica si está marcada la opción <i>Show Animation</i> del menú <i>Run</i> .
AntitheticRandom-Variates	Si su valor es verdadero, las funciones de números aleatorios de Extend generan números aleatorios antitéticos.
CurrentSim	Número de la simulación actual.
CurrentSense	Usado por el análisis de sensibilidad para determinar el número de la simulación para realizar el cambio de variables de sensibilidad.
CurrentStep	Número del paso actual.
CurrentTime	Tiempo actual en la simulación.
DeltaTime	Incremento de tiempo por paso. Sólo para simulaciones continuas.
EndTime	Tiempo, especificado en la configuración de la ejecución, en el que finaliza la simulación.
GlobalProofCmd	Usado por <i>Proof Animation</i> .
MaxAttribs	Permite que un número variable de atributos sea configurado.
ModernRandom	Informa el estado del generador de números aleatorios.
NumSims	Número de veces que la simulación será repetida. Configurado en la simulación.
NumSteps	Número de pasos que serán ejecutados en una simulación continua.
RandomSeed	Número utilizado para generar números aleatorios. Se configurado en la simulación.
SimDelay	Informa el método de orden de simulación.
SimMode	Modo de simulación. Se determina en el cuadro de diálogo de configuración de la simulación.
StartTime	Tiempo de inicio de la simulación. Se determina en el cuadro de diálogo de la configuración de la simulación.

Tabla 10.5: Variables de sistema

- 10.4. Variables globales

Existen tres conjuntos de variables globales que pueden usarse en cualquier parte del modelo. Son útiles para pasar información entre los bloques. Extend no modifica sus valores, excepto cuando se inicia la ejecución para inicializarlas, por lo tanto pueden ser utilizadas libremente por el programador. Los nombres de estas variables comienzan con Global.

Existe otro conjunto de variables globales que son utilizadas por las librerías incluidas en Extend. Sus nombres comienzan con SysGlobal, y su uso está reservado para estas librerías.

Nombre	Tipo	Uso
Global0 a Global9	Real	General
GlobalInt0 a GlobalInt9	Integer	General
GlobalStr0 a GlobalStr9	String	General
SysGlobal0 a SysGlobal9	Real	Reservado
SysGlobalInt0 a SysGlobalInt9	Integer	Reservado
SysGlobalStr0 a SysGlobalStr9	String	Reservado

Tabla 10.6: Variables globales

- **10.5. Vectores globales**

Los vectores globales provee un repositorio, o base de datos interna, para datos específicos del modelo. Son vectores dinámicos que, al igual que las variables globales, pueden ser accedidos por cualquier bloque del modelo.

- **10.6. Listas encadenadas**

Las listas encadenadas son estructuras similares a colas que mantienen punteros entre sus elementos. Esto les permite acelerar el movimiento de elementos dentro de la lista. Sus principales características son:

- Cada elemento puede contener simultáneamente números enteros, reales, cadenas de caracteres o estructuras de datos complejas.
- Son más rápidas para ordenar.
- Pertenecen a bloques individuales pero pueden ser accedidas desde cualquier bloque.

- **10.7. Constantes**

ModL provee cuatro constantes predefinidas:

PI = 3.14159265358;

BLANK = (noValue);

TRUE = 1;

FALSE = 0;

BLANK se utiliza para poner en blanco un ítem de un cuadro de diálogo.

- **10.8. Funciones**

ModL provee un extenso conjunto de funciones. Su enumeración excede el alcance de este trabajo. Se describirán los tipos básicos de funciones y las categorías en las que se clasifican las funciones provistas.

Tipo de función	Categoría
Matemáticas	Matemáticas básicas, Trigonometría, Números complejos, Estadísticas y distribuciones aleatorias, Financieras, Integración, Matrices, Manejo de bits, Ecuaciones.
Entrada/Salida	Entrada/Salida a archivo, Acceso a Internet, Comunicación entre procesos (IPC), OLE/COM, ODBC, Entrada/Salida Serial, Otros drivers, DLLs, Alertas, Entrada de usuario.
Animación	Animación
Bloques y comunicación entre bloques	Números/labels/nombres/tipos/posición de bloques, Información de conexiones, Ítems de diálogo de otros bloques, Cuadros de diálogo de bloques, Tablas de datos de bloques, Mensajes a bloques, Íconos.
Reportes	Reportes.
Plotters	Plotters.
Vectores, colas, listas	Vectores dinámicos, Paso de vectores, Vectores globales, Colas, Listas
Scripts	Construir y ejecutar un modelo remotamente.
Misceláneas	Cadenas de caracteres, Atributos, Fecha/Hora, Unidades de tiempo, Depuración, Ayuda, Modelos y Notebooks, Plataformas y versiones.

Tabla 10.7: Tipos y categorías de funciones

11. Anexo E: Bibliografía

- Churchman, C. W.: «An analysis of the concept of simulation», South-Western Publishing (1963).
- Gordon, G.: «Simulación de Sistemas», 1ra. edición, Diana (1980).
- Imagine That, Inc.: «Extend v6 User's Manual» (2002).
- Imagine That Inc.: «Extend v6 Developer's Reference» (2002).
- Law, A. M., Kelton, W. D.: «Simulation Modeling and Analysis», 2da. edición, McGraw-Hill (1991).
- Lawrence, J. A., Pasternack, B. A.: «Applied Management Science», 2da. edición, Wiley (2002).
- Naylor, Balintfy, Burdick, Kong Chu: «Técnicas de Simulación en Computadoras», 1ra. edición, Limusa (1980).
- Neto, C., Leal, J., Castro, J., García, E., Guerrero, A.: «Validación de Modelos de Simulación» (1998).
- Shannon, R.: «Simulación de Sistemas: Diseño, Desarrollo e Implantación», 1ra. edición, Trillas (1988).
- Shubik, M.: «Simulation of the industry and the firm», American Economic Review (1960).
- Teichroew, D.: «A history of distribution sampling prior to the era of the computer and its relevance to simulation», American Statistical Association (1965).
- <http://imaginethatinc.communityzero.com/extend>. Extend E-Xchange. Virtual Meeting Room for Simulationists.
- <http://www.hps-inc.com/>. Sitio Web de ithink.
- <http://www.imaginethatinc.com/>. Sitio Web de Imagine That Inc.
- <http://www.lionhrtpub.com/orms/surveys/Simulation/Simulation.html>. OR/MS Today – August 2003 – Simulation Software Survey.
- <http://www.powersim.no/>. Sitio Web de Powersim.
- <http://www.sytsa.com>. S&T S.A. Consultores en Alta Dirección.

Notas

1. Churchman, C. W.: «An analysis of the concept of simulation», South-Western Publishing (1963).
2. Shubik, M.: «Simulation of the industry and the firm», American Economic Review (1960).
3. FIFO: «First In First Out». El primero que ingresa es el primero que egresa.
4. LIFO: «Last In First Out». El último que ingresa es el primero que egresa.
5. Teichroew, D.: «A history of distribution sampling prior to the era of the computer and its relevance to simulation», American Statistical Association (1965).
6. Lawrence, J. A., Pasternack, B. A.: Applied Management Science, 2da. edición, Wiley (2002).
7. Schruben, L.W.: «Establishing the credibility of simulations», Simulation (1980).
8. El Vocabulario Científico y Técnico de la Real Academia de Ciencias Exactas, Físicas y Naturales, Tercera edición, Madrid, 1966, define la robustez como: Característica de ciertos procedimientos estadísticos que se muestran prácticamente insensibles a posibles variaciones del modelo probabilístico supuesto. (pág.892)
9. Costos en dólares americanos
10. FCFS: «First Come, First Served». Se atiende primero al que llega primero
11. Porcentaje del tiempo en el que hubo ítems en la cola durante la simulación.
12. Tabla extraída y traducida de OR/MS Today – August 2003