

Capítulo 7 Lenguajes de propósito General, lenguajes de Simulación y Simuladores

7.1 Introducción

La masiva utilización de la informática en la enseñanza y en el entorno industrial, la sorprendente y revolucionaria evolución de los computadores personales en cuanto a tamaño, costo, velocidad, softwares, etc. han ayudado sin lugar a dudas a que la simulación digital o simulación por computadora sea hoy en día la herramienta más utilizada para realizar experimentos de simulación de sistemas. Un programa de simulación de computadora se puede definir como una secuencia de instrucciones que el usuario define para resolver un problema que puede estar plasmado en unas ecuaciones que describen a un sistema que previamente hemos modelizado mediante dichas ecuaciones.

La construcción de un modelo de simulación ha pasado, de ser una labor reservada a especialistas en programación, de difícil y costosa realización, basada en procesos de lotes y en una interpretación en general elaborada a partir del procesamiento de tediosos listados, a ser un ejercicio estructurado alrededor de la utilización de entornos cada vez mas amables y flexibles que permiten aprovechar la característica mas destacable de la simulación : la posibilidad de estudiar la evolución dinámica de los sistemas a lo largo del tiempo.

Hoy en día al ingeniero se le abren un amplio abanico de posibilidades para resolver estos problemas y para programas estas operaciones necesarias para realizar la simulación. El abanico corresponde a los distintos lenguajes que podemos utilizar para traducir nuestros modelos en un computador y posteriormente resolverlos para obtener la simulación del comportamiento del sistema modelado. Podemos utilizar lenguajes de programación general, lenguajes específicos para simulación (Lenguajes de propósito especial) o paquetes de software de simulación especialmente preparados para la misma. Aunque se han utilizado para realizar el ejercicio de la simulación ciertas herramientas como el EXCELL y Paquetes Integrados de Métodos Cuantitativos para la Toma de Decisiones, estos la limitan en su alcance.

A la hora de elegir una herramienta u otra hay que tener en cuenta primeramente la velocidad de ejecución de los programas y la utilización de recursos necesaria (memoria, coprocesadores, etc.).

Hay distintos niveles de lenguajes, en el más bajo nivel se encuentra el lenguaje máquina cuyas instrucciones se escriben en la notación binaria que corresponden directamente con las funciones u operaciones elementales. Este lenguaje es sin duda el más tedioso y menos práctico de utilizar. En un nivel superior se encuentran el lenguaje ensamblador que utiliza símbolos (caracteres) nemónicos para representar dichas funciones.

Los lenguajes de alto nivel o lenguajes de propósito general tales como C, Fortran, Basic, Cobol, Lisp, Algol, Pascal, etc. normalmente alejan al programador de las tareas de bajo nivel de la computadora y suelen ir apoyados en un conjunto de librerías que en el caso de la simulación facilitan mucho la tarea de modelizar los sistemas y reducen normalmente el tiempo de ejecución del programa.

En los años sesenta se realizaban estudios de simulación cuyos costos se medían en años-hombre y su duración en meses. En los setenta aparecieron diversos lenguajes específicamente orientados a la simulación tales como SIMSCRIPT, etc. La década de los ochenta supuso la adaptación sobre PC de productos ya existentes y la aparición de nuevos productos como SIMAN.

Los noventa han protagonizado hasta ahora una auténtica explosión de nuevos productos de

manejo más intuitivo bajo entornos gráficos como Windows. Es el caso de Simfactory, ProModel, Witness, Arena, Taylor II o Simvox, por ejemplo. La evolución de los computadoras y del software comercial se dirige hacia sistemas que puedan ser manejados por personas no-especialistas, con máquinas cada vez más potentes a menor coste. Las técnicas orientadas al objeto conducen a programas de utilización más intuitiva. Todo ello nos sugiere un incremento considerable de la aplicación de las técnicas de simulación.

Sin embargo, a pesar de todo, se estima que en el mercado norteamericano, que es el más desarrollado, sólo se tiene en cuenta la aplicación de técnicas de simulación en un 30% de los casos en los que podría aplicarse, y de este porcentaje, sólo en el 10% de los casos se utiliza regularmente. En Europa las cifras son menores, situándose en torno al 3%, a excepción de Inglaterra donde dicho porcentaje se eleva al 15%.

En el caso de utilizarse un lenguaje específico de simulación, la limitación está en que no permite desarrollar más allá de para lo que está pensado y diseñado el software, pero como contrapartida está que el usuario sólo precisa disponer de los conocimientos de programación relativos al producto. Los productos de modelización visual permiten realizar prototipos en tiempos récord siempre que los objetos a utilizar coincidan exactamente con los disponibles en el producto. En la medida que se requieran objetos específicos hay que recurrir a la programación.

La **Simulación Visual Interactiva**, que puede definirse como aquella que posibilita la creación gráfica de modelos de simulación, permite mostrar por pantalla dinámicamente el sistema simulado, así como la interacción entre el usuario y el programa en ejecución. La interacción implica que o bien se detiene la simulación y solicita información al usuario, o bien que éste puede parar la simulación a su voluntad e interactuar con el mencionado programa; esto último se puede realizar "off-line" o "on-line", es decir sin interrumpir la simulación, e introduciendo las variaciones oportunas tanto en los modelos, como en los valores de las variables en el siguiente ciclo de scan del proceso de ejecución del programa en la computadora que para esto debe tener una estructura multitarea que permita este tipo de operaciones. Algunos productos del mercado son: SIMFACTORY DE CACI Inc., PROMODEL de ProModel Corporation, ARENA de Rockwell

Software Inc., WITNESS de ATT & Istel , o FACTOR/AIM de Pritsker Corporation , FIX DEMACS de Intellution (Fisher-Rosemount). Todos ellos son productos orientados primordialmente a la utilización de la simulación para la resolución de problemas en el ámbito de la producción. Utilizables desde entorno Windows, y ejecutables sobre computadoras personales o sobre plataformas mas potentes como Estaciones de trabajo (Workstations).

Estos permiten construir modelos complejos de manera incremental, a partir de la selección de componentes del sistema de entre un repertorio limitado a la extensión de las librerías que contienen unas entidades predefinidas, si bien las ultimas tendencias añaden a estos paquetes editores para crear nuevas plantillas con características a gusto del consumidor, introduciendo además utilidades de todo tipo incluidas las gestiones de configuración y control de las comunicaciones con un sistema de control real al que se puede conectar el equipo.

Ventajas:

- Sirven para comunicar la esencia del modelo de simulación a los directivos.
- Puede ayudar a corregir errores del programa de simulación, o a mostrar que el modelo no es válido.
- Puede ayudar a entender el comportamiento dinámico del sistema.

Inconvenientes:

- No puede sustituir a un cuidadoso análisis estadístico de los resultados.
- Sólo una parte de la lógica del modelo de simulación puede verse en la animación, y no se puede concluir a partir de ese corto periodo de tiempo que el modelo está bien definido.
- Aumenta el tiempo para desarrollar el programa de simulación.
- Muy lenta la animación en directo.

Otro enfoque se puede derivar de los lenguajes de simulación y de los simuladores es el de los Sistemas Híbridos que combinan la flexibilidad de un lenguaje de simulación con la facilidad de uso de un simulador como lo son el ARENA y el QUEST.

Los simuladores y lenguajes de simulación pueden adoptar uno de los diferentes métodos o estrategias. Existen tres estrategias que son generalmente reconocidas:

- **Enfoque de modelado basado en eventos.** La orientación basada en Eventos (ES) es gobernada por un calendario y ejecución de subrutinas (eventos) que como consecuencia programa la ejecución de otras subrutinas. Los eventos son los instantes de tiempo en los cuales un cambio en el sistema ocurre y coincide con el inicio o terminación de las actividades. Bajo este enfoque segmentos del programa son empleados para definir cada evento en el modelo. Después de inicializado el modelo, las rutinas de ejecución revisan los tiempos de ocurrencia de los eventos y avanzan el reloj de la simulación hacia el tiempo en el cual ocurrirá el próximo evento. Debe existir una subrutina para cada tipo de evento,
- **Enfoque de modelado basado en actividades.** La orientación basada en Seguimiento de Actividades (SA) bajo este enfoque un segmento del programa es empleado para definir cada actividad en la cual las entidades se ven involucradas y las condiciones bajo las cuales la actividad puede realizarse. Dicho segmento incluye una serie de pruebas para determinar si la actividad ha sido iniciada en un punto del tiempo y define las acciones que se deben ejecutar si la actividad ha sido iniciada,

- **Enfoque de modelado basado en procesos.** La orientación basada en Interacción de Procesos (IP) es desarrollada desde el punto de vista de las entidades (transacciones) que fluyen en el sistema. Bajo este enfoque las entidades se clasifican en transacciones o clientes, servidores o recursos (entidades permanentes y entidades temporales). En este enfoque, existen segmentos del programa que son empleados para describir los procesos en los cuales se ven involucradas las entidades

7.2 Simulación en Hoja Electrónica de cálculo y Paquetes Integrados.

La hoja electrónica es una plataforma de modelación viable pero tiene sus limitaciones, en amplitud del conocimiento técnico requerido y en el tamaño de los modelos aceptados. Los usuarios expertos pueden extender de forma significativa las capacidades de la hoja electrónica (como Excel) con un lenguaje de programación de propósito general (como Visual Basic VBA), pero tales tópicos generalmente caen fuera del alcance de los cursos cuantitativos. Generalmente en un curso el estudiante permanece novicio en las herramientas cualitativas para la administración. Por esto, un curso de reconocimiento deberá proveer el conocimiento básico del rango de técnicas, una comprensión suficiente de las características más relevantes de la técnica y lo más importante, una apreciación de que es aún no familiar con respecto a la técnica.

La simulación analítica cae en dos categorías:

- Muestreo o Simulación de Monte Carlo, y
- La simulación de procesos (o simulación de eventos discretos).

Evans (2001) describe estos como simulación de Monte Carlo o simulación con riesgo y sistemas de simulación, respectivamente.

Las aplicaciones del Monte Carlo en simulación son perfectas para simularse en una hoja electrónica, son usadas para examinar el riesgo o incertidumbre asociada con modelos estáticos o simulaciones que incluyen enfoques de búsqueda de actividades. Pueden ser analizados en la hoja electrónica algunos modelos típicos como son: modelos de inversión, modelos de inventarios, modelos sencillos de líneas de espera. (Ragsdale, [2001](#); Camm y Evans, [1996](#)). La hoja electrónica fácilmente recalcula un modelo, sus funciones proveen generadores de números aleatorios, y tiene capacidad plena de grabar y analizar los datos generados de la simulación. Cercanamente todos los textos incluyen un capítulo sobre la simulación, un capítulo generalmente enfocado a la simulación de Monte Carlo. Adiciones como **@Risk y Crystal Ball, io_jensen_excel, y MS Courseware de Hillier y Lieberman** incrementan significativamente la habilidad del usuario para conducir, y enseñar método de Monte Carlo en simulación, también como realizar análisis de riesgo en estudios de simulación o proyectos estudiantiles y tesis.

La simulación de procesos no es del todo aplicable para las hojas electrónicas. La simulación de procesos es usada para capturar cambios de estado de los sistemas complejos con respecto al tiempo, particularmente cuando los cambios de estado son definidos por eventos dentro del sistema.

Las interacciones dinámicas y el orden incierto de los eventos dificultan para capturar un modelo en una hoja electrónica sin de alguna manera aumentar las capacidades de la misma. Ejemplos típicos pueden incluir operaciones de mantenimiento incluyendo demandas sobre recursos limitados y eventos no predecibles de falla, procesos de producción incluyendo los ciclos dentro de estaciones de trabajo en una instalación, o aún modelos complejos de combate con sistemas interactivos múltiples envueltos en un conflicto. Estas aplicaciones más complicadas y con

frecuencia mas realistas de la simulación no son discutidas o muy brevemente mencionadas cuando se hace referencia a paquetes de simulación mas sofisticados. Excepciones se incluyen Camm y Evans (1996), Evans y Olsen (1998), y Laurence y Pasternack (1998) cuyos trabajos incluyen versiones estudiantiles de esos poderosos paquetes. Estas excepciones son, sin embargo admitidamente limitadas en el tiempo y espacio dedicado a discutir como usar esos poderosos paquetes de simulación.

Una visión de fracaso de la simulación debido a las limitaciones de la hoja electrónica se observa cuando se tiene una completa apreciación del poder y los beneficios de la simulación como una técnica cuantitativa. Para subsanar esta limitación, existen algunos paquetes como el SimQuick.

SimQuick (Hartvigsen, 2001) es un paquete de simulación orientado al proceso para excel. SimQuick no es una adición para Excel (tal como un archivo .xla), en su lugar es una plantilla de hoja electrónica en la cual el usuario especifica los elementos de una simulación, la parametrización de estos elementos, y la conexión entre los elementos. Estas especificaciones son indicadas a el SimQuick via tablas de una hoja electrónica particular cuyo acceso esta controlado vía menú del VBA dentro de la hoja electrónica del SimQuick, el SimQuick puede ser descrito como un paquete de primera generación, repleto de limitaciones importantes cuando es comparado con paquetes maduros de propósito especial (simuladores) tales como Extend, Witness, Taylor Ed, ProModel, o Arena. Si embargo, encontramos que el SimQuick, a pesar de sus limitaciones, es muy adecuado para proveer una herramienta de hoja electrónica con la cual enfocarse en las características de salida del proceso de simulación, además de proveer una cobertura completa de la simulación como una técnica cuantitativa.

Algunos paquetes de métodos Cuantitativos para la toma de decisiones incorporan módulos en sus menús para resolver problemas de poca complejidad que requieren el uso de la simulación. Entre estos tenemos el SIMNET II incluido en el TORA del libro de Investigación de Operaciones, Hamdy A. Taha, Editorial Mcmillan y el QSIM incluido en el WinQSB, Versión 2.0 por Yih-Long Chang, Kiran Desai, editorial Wiley.

Página de la cual puedes bajar el WinWSB <http://personalpuntocom.tripod.com.pe/winqsb.html>

SIMNET II

SimTec, Inc., P.O. Box 2492, Fayetteville, AR 72702

Phone:(479)756-6146

FAX:(479)75-7446

email: hat@enr.uark.edu

<http://web.iner.uark.edu/Fsimnet/Fsimnet.htm>

7.3 Lenguajes de propósito general

El Fortran es un lenguaje científico de programación y no fue específicamente diseñado para ser usado en la simulación. Cuando se usa el FORTRAN el analista probablemente adopte la orientación de eventos programados.

Actualmente, C representa "el lenguaje de programación", debido a que está orientado cien por ciento hacia los programadores (a diferencia de otros lenguajes, que aunque también de programación, originalmente fueron creados con una orientación especial; como el caso de COBOL: un lenguaje orientado hacia los negocios; o bien, Pascal, cuyo propósito original fue ser un lenguaje didáctico); C es un lenguaje compacto: cuenta sólo con alrededor de 30 palabras clave órdenes que constituyen el lenguaje-, que presenta pocas restricciones y quejas (señalización de errores), y genera un código rápido y eficaz,

C constituye un lenguaje de nivel medio, esto, en oposición a lo que pudiere pensarse, significa mayor poder, eficacia y flexibilidad; ofrece las ventajas de los lenguajes de alto nivel como Pascal o ADA (además puede considerarse como un lenguaje estructurado), así como proporcionar la eficiencia del lenguaje ensamblador, al permitir la manipulación directa de bits y bytes, sin (os problemas asociados al desarrollo y depuración de código en ensamblador.

Otra importante característica de C es su gran portabilidad, lo cual significa que el código escrito en este lenguaje puede funcionar, prácticamente, sin cambio alguno, en los más diversos equipos de cómputo y sistemas operativos, desde microcomputadoras (Pc's y Apple, etc.) que utilizan sistemas operativos como DOS, Mac OS, Windows, y Linux, hasta los equipos mini, mainframes y supercomputadoras con sistemas como UNN, MVS y VM/ESA, Por cierto, UNIX, uno de los más importantes sistemas operativos de hoy en día está escrito, casi en su totalidad, en C.

Dadas sus características y posibilidades, C puede denominarse como un lenguaje de propósito general, debido a que su utilización permite el desarrollo de las más diversas aplicaciones de software, Concebido originalmente como un lenguaje para la creación de software de sistema (sistemas operativos, compiladores, editores, administradores de bases de datos, etc.), el lenguaje C es ideal para la implementación de cualquier tipo de sistemas y programas sin importar su giro o magnitud (software administrativo, científico, de investigación, educativo, de control interno, de entretenimiento y cualquier otro tipo de aplicación o problema que se tenga en mente),

Además, es importante mencionar que C ha sido retornado como base para la generación de nuevos lenguajes; tal es el caso del lenguaje C++, al cual podemos pensar como la versión orientada a objetos de C. C++ es considerado el sucesor de C, ya que incorpora a las ventajas de éste último, las características del paradigma de la programación orientada a objetos.

Así, C++ constituye lo que se denomina el súper conjunto de C orientado a objetos, de modo que al usar C++ el programador dispone de, prácticamente, todas las características y posibilidades de C, más las características nuevas. Además, C++ implementa el concepto de herencia múltiple, el cual no está permitido por otros lenguajes de programación orientados a objetos.

En relación a la programación para sistemas de control gráfico, como Windows, se han incorporado a C++ librerías de clases para la generación y administración de objetos gráficos (ventanas, botones, barras de desplazamiento, iconos, etc.), entre las cuales destacan la Microsoft Foundation Classes (MFC) y las librerías Object Window Library (owl) desarrolladas por Borland Software Corporation. Mediante éstas se facilita el desarrollo de las complicadas aplicaciones para el

ambiente Windows, ya que el basarse en propiedades de los programas de orientación a objetos, les permite un diseño más claro y lógico de los mismos.

Anteriormente las aplicaciones para Windows se desarrollaban en C empleando un enfoque funcional basado en intercambio de mensajes más complejo, determinado en el Microsoft Developers Kit (MDK): serie de funciones y estructuras de datos que permiten la generación y control de elementos gráficos.

Igualmente, la programación de aplicaciones en ambientes distribuidos (redes Intra e incluso internet), es posible a través de C++; sin embargo, dadas las características de construcción interna de éste, es necesario para tales aplicaciones tener un especial cuidado en cuanto al manejo de aspectos de seguridad.

Otro de los aspectos más atractivos de C++, es que ha sido empleado como base para el desarrollo de gran parte de los programas que soportan los conceptos de la tecnología de realidad virtual: la representación y simulación de nuestro entorno en el espacio binario.

Características de los lenguajes de propósito general (FORTRAN, C++, Pascal, Visual Basic, Visual C)

- Tedioso, complejo, bajo nivel
- Muy flexibles
- Flexibilidad limitada — validación del modelo?
- Tediosos, de bajo nivel, sujetos a cometer errores
- Pero, con una gran flexibilidad
- “Paquetes” de soporte
- Consisten en subrutinas para procesamiento de listas, contabilizar,
- Avanzar el tiempo
- Difusión amplia. Fáciles de modificar.
- Hojas de cálculo
- Generalmente: modelos estáticos
- Análisis de escenarios, muestreo de distribuciones, Control de calidad.

De todas formas, frente a la disyuntiva de tener que elegir uno u otro tipo de programación, lo primero a tener en cuenta es que la programación con un lenguaje de propósito general como C++, apoyado con librerías, permite afrontar problemas de la máxima complejidad y tamaño pudiendo ser el programa altamente movible. El mayor inconveniente de este tipo de programación se presenta al interpretar los mensajes de error del sistema de desarrollo, puesto que se requiere un alto grado de especialización en el desarrollo de software. (Otros lenguajes de programación de propósito general pudieron ser discutidos, pero no es la intención de este libro.)

A continuación se analizan las características de FORTRAN como un lenguaje de propósito general usado para llevar a cabo la práctica de la simulación:

7.3.1 Simulación en FORTRAN

El FORTRAN es un lenguaje de programación ampliamente conocido y disponible., que se ha usado extensivamente en la simulación. Sin embargo, no provee los elementos directamente dirigidos para ayudar al analista de simulación quién es forzado a programar todos los detalles de un algoritmo avanzado de programación al orientada al evento, la capacidad de obtención de estadísticas, la generación de muestras de distribuciones de probabilidad específicas, y un generador de reportes. (Sin embargo, varias librerías de subrutinas científicas, tales como IMSL, contienen numerosos generadores de variables aleatorias.) Para modelos grandes, el uso del FORTRAN puede resultar incomodo; adicionalmente, puede resultar en modelos los cuales son difíciles de depurar u ejecutar de forma lenta, a menos que sean usado un enfoque cuidadoso, organizado y una lista de técnicas de procesamiento efectivo. Para modelos pequeños, la simulación con FORTRAN (o cualquier otro lenguaje de propósito general) puede usarse como una herramienta de aprendizaje para reforzar los conceptos del algoritmo avanzado de programación orientada al evento. Lo mas importante es que el lenguaje de propósito general esconde los detalles de la programación orientada al evento.

Cualquier modelo de simulación de eventos discretos escrito en FORTRAN contiene los componentes: estado del sistema, entidades y atributos, eventos, actividades, y retrasos, además de los componentes listados a continuación. Los siguientes componentes son comunes para casi todos los modelos escritos en FORTRAN:

- **CLOCK** una variable que define el tiempo de simulación
- **Subrutina de inicialización.** Una rutina para definir el sistema en el tiempo 0
- **Subrutina del avance del tiempo.** Una rutina que busca la lista de eventos futuros (FEL) par en evento próximo (llamado el evento inminente, IMEVT) y avanza el reloj a el tiempo de ocurrencia del evento inminente.
- **Subrutina de programación.** Una rutina que coloca los eventos futuros generados en el FEL
- **Subrutinas del evento.** Para cada tipo de evento, una subrutina actualiza el estado del sistema (y sus estadísticas acumuladas) cuando ocurra el evento.
- **Generador de variables aleatorias.** Rutina para generar muestras de la distribución de probabilidad deseada.
- **Programa principal.** Provee el control general del algoritmo de programación del evento.
- **Generador de reportes.** Una rutina que calcula el resumen de estadísticas de las estadísticas acumuladas e imprime un reporte al final de la simulación.

7.4 LENGUAJES DE SIMULACIÓN (Lenguajes de propósito especial)

7.4.1 Introducción

Los lenguajes de simulación facilitan enormemente el desarrollo y ejecución de simulaciones de sistemas complejos del mundo real. Los lenguajes de simulación son similares a los lenguajes de programación de alto nivel pero están especialmente preparados para determinadas aplicaciones de la simulación. Así suelen venir acompañados de una metodología de programación apoyada por un sistema de símbolos propios para la descripción del modelo por ejemplo mediante diagramas de flujo u otras herramientas que simplifican notablemente la modelización y facilitan la posterior depuración del modelo.

Características de los lenguajes de simulación:

- Los lenguajes de simulación proporcionan automáticamente las características necesarias para la programación de un modelo de simulación, lo que redundará en una reducción significativa del esfuerzo requerido para programar el modelo.
- Proporcionan un marco de trabajo natural para el uso de modelos de simulación. Los bloques básicos de construcción del lenguaje son mucho más afines a los propósitos de la simulación que los de un lenguaje de tipo general.
- Los modelos de simulación son mucho más fácilmente modificables.
- Proporcionan muchos de ellos una asignación dinámica de memoria durante la ejecución.
- Facilitan una mejor detección de los errores.
- Los paquetes de software especialmente diseñados para simulación contienen aplicaciones diversas que facilitan al simulador las tareas de comunicaciones, la depuración de errores sintácticos y de otro tipo de errores, la generación de escenarios, la manipulación "on-line" de los modelos, etc.
- Son muy conocidos y en uso actualmente
- Aprendizaje lleva cierto tiempo
- Simuladores de alto nivel
- Muy fáciles de usar por su interfase gráfica
- Restringidos a las áreas de manufactura y comunicaciones
- Flexibilidad restringida puede afectar la validez del modelo

Entre estos lenguajes específicos podemos nombrar los siguientes:

MIDAS, DYSAC, DSL , GASP, MIMIC, DYNAMO, GPSS, SIMULA, CSSL(Continuous System Simulation Language) , CSMP, ACSL (Advanced Continuous Simulation Language), DARE-P and DARE-Interactive, C-Simscript, SLAM, SIMAN, SIMNON, SIMSCRIPT-II-5, ADA, GASP IV, SDL. Muchos de estos lenguajes dependen fuertemente de los lenguajes de propósito general como es el caso de SLAM o SIMAN que dependen de Fortran para las subrutinas.

Por otro lado, el GPSS es un caso especial de un lenguaje de simulación de propósito especial, altamente estructurado que está orientado a la transacción, un caso especial de una orientación basada en procesos más general. El GPSS fue diseñado para la simulación simple de sistemas de colas tales como trabajos de taller. A diferencia de los otros lenguajes de simulación, GPSS tiene varias implementaciones incluyendo GPSS/H y GPSS/PC, ambos de los cuales serán discutidos más adelante.

El SIMAN V, SIMSCRIPT II.5, y el SLAM son lenguajes de simulación de alto nivel que tienen constructor especialmente diseñados para facilitar la construcción de modelos. Estos lenguajes proveen al analista de simulación con una opción orientación basada en procesos o basada en eventos, o un modelo usando una mezcla de las dos orientaciones. A diferencia del FORTRAN, estos tres lenguajes proveen la administración de la lista de eventos futuros, generador interno de variables aleatorias, y rutinas internas para la obtención de estadísticas (estas características para las implementaciones del GPSS mencionadas previamente.) Se pueden lograr cálculos complejos en ambas implementaciones del GPSS y estos tres lenguajes. El SIMAN, SIMSCRIPT II.5, y el SLAMSYSTEM proveen la capacidad de realizar simulación continua (esto es, para modelar sistemas que tengan continuamente cambios en sus variables de estado) pero este tema no está dentro del alcance de este libro.

El SIMAN está escrito en C, aunque las primeras versiones del lenguaje fueron escritas en FORTRAN. El SIMAN V puede ser ejecutado directamente, o a través del medio ambiente del ARENA.

El SLAMSYSTEM contiene al lenguaje de simulación SLAM II. El SLAM II está basado en el FORTRAN y contiene al lenguaje GASP como un subconjunto. El GASP es un conjunto de subrutinas en FORTRAN para facilitar las simulaciones orientadas al objeto escritas en FORTRAN. El SIMSCRIPT II.5 por otro lado, contiene un subconjunto de un completo lenguaje científico de simulación comparable con el FORTRAN, C o C++.

El MODSIM III es un descendiente del lenguaje que la compañía de productos CACI originalmente diseñó por la armada de los Estados Unidos. Hereda mucha de su sintaxis del MODULA-2 y del ADA, ciertas características del ADA y sus conceptos de simulación del SIMSCRIPT y el SIMULA. Algunas de las características de la simulación orientada al objeto fueron originalmente vistas en el SIMULA y el SMALLTALK.

A continuación se muestra un análisis de las características principales de los principales lenguajes de simulación **GPSS, SIMAN, SIMSCRIPT II.5, SLAM II usando SLAMSYSTEM, y el MODSIM III.**

7.4.2 Simulación en GPSS

El GPSS es un lenguaje altamente estructurado, un lenguaje de simulación de propósito especial que usa el enfoque basado en procesos y se orienta hacia los sistemas de colas. Un diagrama de bloques provee una forma conveniente para describir el sistema que se está simulando. (Existen más de 40 bloques estándar en el GPSS). Las entidades llamadas transacciones pueden ser vistas como que fluyen a través de un diagrama de bloques. Por lo anterior, GPSS puede ser usado para modelar una situación donde las transacciones (entidades, clientes, unidades de tráfico) están fluyendo a través del sistema (ejem; una red de líneas de espera, con las colas precediendo a recursos escasos). El diagrama de bloques es preparado en una forma que reconozca la

computadora junto con los estatutos de control para que simulación sea desarrollada por el procesador.

El GPSS fue liberado por IBM en 1961. La implementación original ha sido nuevamente implementada y mejorada en muchas partes desde 1961, dos de esas implementaciones son GPSS/H y GPSS/World.

EL GPSS/H es un producto de Wolverine Software Corporation, Annandales, VA. EUA. Es una herramienta flexible y poderosa de simulación. Las mejoras mas allá de la implementación del GPSS incluyen un archivo predefinido de entrada y salida de información, el uso de expresiones aritméticas como operadores de bloque, depurador interactivo, ejecución más rápida, disponibilidad de estatutos de controles expandidos, y variables ampliadas que permitan la combinación de valores aritméticos usados en simulación. La última liberación del GPSS/H es la versión 2.0. Añade un reloj de punto flotante, funciones matemáticas predefinidas, y generadores de variables aleatorias predefinidos. Las opciones disponibles incluyen el GPSS/H personal con un límite de memoria de 640K , y GPSS/H 386 que permite un tamaño de modelo ilimitado.

Otro producto de Wolverine Software Corporation, Annandales, VA. EUA, El animador para el GPSS/H se llamado Prof. Animation. Cualquier software puede escribir datos ASCII a un archivo puede conducir pruebas de animación. Por lo anterior, además del GPSS/H, BASIC, C++, FORTRAN, SIMAN, SIMSCRIPT II.5, y SLAMSYSTEM pueden servir como conductores. La animación es lograda usando un fondo estático, un archivo de distribución, y un archivo de rastreo que contiene eventos dinámicos.

El GPSS World es un completo rediseño de su predecesor GPSS/PC. Este incluye la simulación discreta y continua. Opera bajo el ambientes OS/2. Sus características incluyen interactividad, visualización, y flexibilidad en la configuración. Utiliza 32 bits, memoria virtual, multitareas y multiprocesamiento simétrico y simulación distribuida.

7.4.3 Simulación en SIMAN

Por sus siglas en ingles SIMAN significa Análisis, modelación y simulación (Simulation Modeling and Análisis). Este lenguaje fue desarrollado por C. Dennis Pedgen, Systems Modeling Corp., Sewickley, PA. EUA. Las capacidades del lenguaje incluyen orientación basada en procesos, orientación basada en eventos, y simulación continua, o una mezcla de cualquiera de dos de los tres enfoques. Se mostrará el primero de los tres enfoques donde las entidades fluyen a través del sistema. El marco de trabajo del SIMAN esta compuesto del modelo y de los marcos experimentales. La actual versión del lenguaje es SIMAN V.

Algunos aspectos importantes del SIMAN V son:

- 1) Características especiales que son útiles en modelar sistemas de manufactura incluyendo la habilidad de describir el medio ambiente de los centros de trabajo (estaciones) y la habilidad de definir una secuencia de entidades en movimiento a través del sistema.
- 2) Constructores que permiten modelar sistemas de manejo de materiales incluyendo bandas de transporte acumulables y no acumulables, transportadores, y vehículos guiados.
- 3) Un controlador de corridas interactivo que permite puntos de cambio, relojes, y otros procedimientos de ejecución.
- 4) El medio ambiente del ARENA que incluye procedimientos de menú dirigido por “apunta y click del mouse” para construir el modelo del SIMAN V y experimentar animación usando CINEMA, el procesador de entradas que asiste en ajustar los datos a distribuciones, y el procesador de salidas que puede ser usado para obtener intervalos de confianza, histogramas, correlogramas, y así sucesivamente.
- 5) La accesibilidad del modelo a todo tipo de computadoras

7.4.4 Simulador SIMSCRIPT II.5

El SIMSCRIPT II.5 de la compañía de productos CACI, es un lenguaje que permite modelar y que puede ser orientado a eventos o orientado a procesos. Las versiones para microcomputadora y estaciones de trabajo incluyen el paquete de animación y gráficos SIMGRAPHICS. El paquete SIMSCRIPT puede ser usado para producir presentaciones gráficas de calidad estáticas y dinámicas tales como Histogramas, graficas de Pastel, Graficas de Barras, gráficas de nivel y gráficas de variables. La animación de la simulación de salida esta también se construye también usando SIMGRAPHICS. Se puede usar para producir graficas interactivas o formas para entrar datos al modelo. Una forma de suministro de datos puede incluir los elementos gráficos como barras de menú con menús desplegables, cajas de texto o datos, y botones para seleccionar con el ratón una alternativa. El modelo gráfico permite para un cierto conjunto de modificaciones en el modelo se realicen sin programación alguna, facilitando el uso del modelo para los que no son programadores.

7.4.5 Simulación en SLAM II usando SLAMSYSTEM

El SLAM II, sacado al mercado por la corporación Pritsker de INDIANA, EUA, es un lenguaje de simulación de alto nivel con versiones de FORTRAN y C. El SLAM II permite una orientación basada en eventos o basado en procesos, o una combinación de ambos enfoques. Esta sección brevemente describe la porción del proceso interacción del SLAM II, y también del SLAMSYSTEM el cual es usado para construir, animar y ejecutar los modelos de simulación SLAM II.

Para usar en enfoque del proceso interacción del SLAM II, el analista de simulación desarrolla una red, consistiendo de nodos y ramas, representando el proceso en un sistema pictórico. Los objetos que fluyen a través del sistema se denominan *entidades*. La red es entonces traducida por el analista en declaraciones reconocidas por la computadora, ya sea usando el SLAMSYSTEM o en formato de texto. Un modelo de red completa de un sistema del SLAM II representa todos los posibles caminos que una entidad puede tomar conforme pasa por el sistema.

El SLAM II automáticamente maneja el algoritmo avanzado de programación de eventos/tiempo, establece operaciones tales como la adición y generación de muestras aleatorias. En SLAM II, los conjuntos son llamados archivos. Con su manejo automático de archivos, el SLAM II puede manejar colas en una base PEPS o UEPS, o las entidades pueden ser categorizados (y servidas) en orden de un atributo tal como una prioridad. El SLAM II tiene una función interna de generación de variables aleatorias para una amplia variedad de distribuciones estadísticas.

Una *rama*, en una red SALM II, representa el tiempo de primer paso; esto es, que representa una actividad. Además, una rama puede ser representada como un número limitado de servidores. Una *rama* es codificada como una actividad. Los nodos son usados para representar el arribo de un evento (nodo CREATE), esperas condicionales o retrasos (nodo de cola), el evento de terminación (nodo TERMINATE), y otras acciones típicas del sistema.

7.4.6 Simulación en MODSIM III

El MODSIM III es un lenguaje de programación de propósito general orientado al objeto. Es un lenguaje compilado que es altamente trasladable. La sintaxis y la estructura esta basada en el MODULA-2. Constructores internos orientados al objeto incluyen herencias sencillas y múltiples, unión dinámica de objetos, polimorfismos, encapsulación, abstracción de datos, y ocultación de información. Este lenguaje permite una interfase a C tal que las librerías existentes en código fuente y objeto de C puede ser incluido en los programas del MODSIM III. El MODSIM III realiza interfase con el animador de CACI SIMGRAPHICS II.

Programación orientada a objetos

Los objetos son estructuras de datos asignadas dinámicamente junto con una rutina, llamadas *métodos*. El campo en la estructura de los datos del objeto define su estado en cualquier instante en el tiempo, mientras que sus métodos describen las acciones que el objeto puede realizar. Los valores de los campos de un objeto pueden ser modificados únicamente por sus propios métodos. Debido que ninguna otra parte del programa puede modificar estos valores, el mantenimiento del programa y la depuración son ampliamente simplificadas.

Otra entidades pueden requerir el valor de los campos de un objeto o preguntar para realizar sus métodos enviando mensajes a el objeto. Esto es una característica importante de los objetos. En lugar de invocar los métodos de los objetos por una llamada, el usuario invoca el método enviando un mensaje al objeto que requiere el método a desarrollar. Este pequeño refinamiento en forma que el código es invocado es el responsable para muchas de las ventajas en la programación orientada a objetos.

7.5 SIMULADORES

7.5.1 Introducción

Son paquetes que permiten simular algunos tipos de sistemas con poca o ninguna necesidad de programar. Los sistemas se seleccionan a base de menús y de gráficos.

- Ventaja: Ahorran tiempo de programación
- Inconveniente: Están limitados a modelizar solamente las configuraciones de sistemas que permite el simulador. Tienen por tanto poca flexibilidad. Si nuestro sistema real no se adapta a ninguna de las configuraciones que tenga programadas el simulador, no deberíamos utilizarlo. Las ejecuciones son más lentas por lo comentado en el apartado anterior.

La simulación de sistemas de manufactura y de manejo de materiales es bastante compleja que es necesario utilizar software especializado que ha sido desarrollado para este propósito. Nueve de los paquetes de software mas utilizados serán descritos a continuación. Los paquetes de software incluyen SIMFACTORY II.5, ProModel, AutoMod, Taylor Ed (FELXYSIM), Witness, AIM, Extend, Arena, y SIMUL8.

7.5.2 Simulador SIMFACTORY II.5

Es un simulador escrito en SIMSCRIPT II.5 y MODSIM III para ingenieros que no son analistas de simulación de tiempo completo. Opera en una PC bajo ambiente Windows y OS/2, y en muchas estaciones de trabajo (Workstations). Un modelo es mejor construido en etapas primeramente definiendo la distribución del proceso que consiste en las estaciones de trabajo, almacenes temporales, áreas de recibo, rutas de transporte, definiendo productos, recursos, y transportadores y finalmente las interrupciones. Una animación basada en Iconos es seguida automáticamente después de definir el modelo. Los elementos del modelo son obtenidos de una tabla en lugar de una barra de menús. El modelo resultante puede ser modificado usando una interfase gráfica o un editor de textos. Se tiene una modelación de flujo flexible. Por ejemplo el lógico OR puede ser usado (como una solicitud de Parte A OR Parte B).

La distribución es creada posicionando Iconos, seleccionados de una librería, o de una pantalla. Conforme cada Icono es posicionado, son dadas las características que los describen. Los productos son definidos por planes de proceso que definen las operaciones desarrolladas por cada parte y la duración de la operación.

Los recursos son añadidos al modelo en dos pasos. Primero, el recurso esta definido y su cantidad y capacidad son establecidas. Segundo, se identifican las estaciones que requieren recursos. Conforme que los recursos se mueven, el tiempo de simulación transcurre. Los requerimientos de los recursos son flexibles, por ejemplo, una unidad del recurso A y dos unidades del recurso B pueden ser requeridos.

Los transportadores pueden ser movilizadotes en lotes, tales como montacargas o pueden ser bandas transportadoras. Son especificadas las características de los transportadores (velocidad de

levantado, tiempo para recoger, tiempo para dejar, y la capacidad de un montacargas). La ruta del transportador es identificada en la pantalla. Los transportadores pueden evitar colisiones y pueden llevar recursos.

Cualquier interrupción, planeada o no, puede ser aplicada a cualquier elemento del modelo o grupo de elementos (ejemplo, Bandas transportadoras, colas, recursos, y transportadores). Las interrupciones pueden requerir cualquier combinación de recursos.

Se tienen reportes disponibles relativos a utilización de equipos, producción, productos en proceso, y utilización de almacenes. Gráficas múltiples pueden ser comparadas al mismo tiempo (pastel, histogramas y barras). Los datos pueden ser comparados a través de múltiples corridas. Los reportes pueden ser arreglados de diversas maneras y exportados a hojas electrónicas, se pueden reunir estadísticas de elementos de interés. Un reporte sumario de todas las réplicas provee medias, desviaciones estándar, e intervalos de confianza del producto del modelo.

United States: CACI International Inc

1100 North Glebe Road Arlington, VA 22201 USA

Tel.: (703) 841-7800

CACI Products Co. 3333 N. Torrey Pines Ct. La Jolla, CA 92037

7.5.3 Simulador AutoMod

El AutoMode de AutoSimulations Inc. Combina las características de los lenguajes de propósito especial (lenguajes de simulación) y un simulador de propósito especial de manejo de materiales. Tiene características generales de programación incluyendo la especificación del proceso y procedimientos del proceso, recursos, cargas, colas, y variables. Los procesos son especificados en términos de límites de tráfico, conexiones de entradas y salidas de sistemas de manejo de materiales, y lógica del proceso. Los recursos son especificados en términos de su capacidad, tiempo de procesamiento, tiempo entre falla y tiempo de reparación.

Las cargas son definidas por su forma y tamaño, sus atributos, tasas de generación, tiempos de inicio, y todas las prioridades.

Como una alternativa para los sistemas del proceso y del lenguaje de propósito general, AutoMod tiene un simulador opcional de trabajos de taller con una interfase similar a una hoja electrónica, en la cual todos los datos y la lógica del modelo pueden suministrarse en un medio ambiente de no programación que es muy similar a una hoja electrónica.

El simulador de manejo de materiales es muy poderoso en la descripción de los sistemas de manejo de materiales. Se pueden definir Vehículos guiados automáticamente AGV's,

transportadores (de banda, canjilones, etc.), grúas de puente, sistemas de almacenamiento y recuperación de materiales AS/RS , sistemas libre de de fuerza. El rango de definición es extensivo. Por ejemplo, un AGV puede ser definido en términos de lo siguiente: tipos de vehículos múltiples, vehículos de capacidad múltiple, rutas opcionales (unidireccional, bidireccional, y de espuela), velocidades, aceleraciones y desaceleraciones basadas en los tipos de carga, puntos de control, control programado y flexible de de rutas, y geometría de bloques arbitrarios, y rutas automáticas de distancia mas corta.

Están disponibles en el proceso del sistema numerosas declaraciones de control, llamadas acciones,. Por ejemplo, las acciones del proceso incluyen if-then-else, while-do, wait-until, y wait-for. El control de la carga, y el control de los recursos son otras acciones también disponibles. Funciones en C pueden ser llamadas en caso necesario, pero no son requeridas en la mayoría de los modelos. También se pueden especificar variables y atributos.

Las capacidades de animación están basadas en dibujos en escala e incluyen vectores gráficos en verdadera 3D, rotación, acercamiento y alejamiento sobre una pantalla virtual en tiempo real. Una utilería de dibujo estilo CAD es usada para construir los elementos gráficos de un modelo. Además, con el producto de adición (add-on) IGES, se pueden importar y convertir dibujos desde sistemas CAD a fondos estáticos o transportadores y rutas. Son generadas gráficas de negocios tales como gráficas de tiempos, gráficas de pastel o de barras.

La última versión 7.5, contiene un simulador dentro del AutoMod para un ambiente trabajos de taller con muchos productos que tienen rutas variables. Las características del simulador incluyen su interfase de hoja electrónica que se conecta al lenguaje de procedimientos del procesos del AutoMod. Mientras que el simulador elimina la necesidad de programación en la construcción de muchos de los modelos, como es la meta de la mayoría de los simuladores, el simulador AutoMod permite las extensiones y aclimatación para aquellas partes de un sistema que no se ajusta a un modelo pre-programado en simulación.

El AutoStat, un paquete estadístico estrechamente integrado con el AutoMod, provee capacidades de calentamiento, escenario gerencial, y otras capacidades estadísticas incluyendo la generación de intervalos de confianza, y diseño de experimentos. Kinematica es otro paquete de adición (add-on), que permite la simulación detallada de robots en 3D, gente y otros objetos con partes en movimiento, ya sea en forma separada o integrada en una simulación mayor de un sistema de manufactura o de manejo de materiales.

El AutoSche, una herramienta para la planeación y programación de la producción de capacidad finita y otros medios ambientes de manufactura discreta de partes, esta basada en el simulador AutoMod y permite una extensiva aclimatación a las reglas de selección par obtener el mejor programa de producción entre las alternativas disponibles.

En el AutoMod, no existe limites con respecto al tamaño del modelo o las capacidades de aclimatación. El AutoMod se puede ejecutar en PC´s y en varias estaciones de trabajo en ambiente UNIX.

Simulation Services Headquarters

*5245 Yeager Road
Salt Lake City
UT 84116-2877
Tel: (801) 736-3201
Fax: (801) 736-3443
E-mail: automod-info@brooks.com*

<http://www.autosim.com/>

Brooks Automation Corporate Headquarters

*15 Elizabeth Drive
Chelmsford, MA 01824 U.S.A.
Tel: 978-262-2400
Fax: 978-262-2500
<http://www.brooks.com>*

7.5.4 Simulador ProModel

ProModel es una herramienta de simulación que funciona en computadoras personales en un ambiente Windows. Mediante una combinación ideal de facilidad de uso, flexibilidad y potencia, permite diseñar y analizar sistemas de producción y servicios de todo tipo y tamaño y modelar prácticamente toda situación, en forma casi real, mediante sus capacidades gráficas y de animación.

ProModel fue concebido como una herramienta para ingenieros y gerentes que desean lograr reducciones de costos, mejoras en la productividad e incrementar las ventajas estratégicas en la producción de bienes y servicios. En resumen, con la simulación se tiene la habilidad para determinar el uso de los recursos disponibles – personal, equipo e instalaciones – mas eficiente y productivamente.

No se necesita que el ingeniero o modelador tenga una gran habilidad para programar. Mediante su interfase gráfica y el uso de pequeños modelos preconstruidos, permite modelar sistemas complejos de producción y servicios en forma fácil y rápida. ProModel por otra parte, se puede utilizar como un medio muy efectivo para probar y generar nuevas ideas de diseño y mejoramiento, antes de realizar las inversiones y/o modificaciones necesarias para construir o mejorar estos sistemas. En la misma forma sirve para identificar cuellos de botella, seleccionar la alternativa que ofrezcan la mejor relación beneficio-costos y hacer Análisis de Sensibilidad (¿Qué pasaría sí?).

Como un simulador de eventos discretos, ProModel esta concebido para modelar sistemas de **manufactura discreta** (unidad por unidad), sin embargo, muchos sistemas de **manufactura continua** pueden ser modelados convirtiendo unidades a granel en unidades discretas tales como galones o barriles. Adicionalmente se puede adaptar fácilmente para modelar sistemas de servicios de salud (Centros de atención medica) o procesos financieros entre otros.

Algunas aplicaciones típicas de ProModel son las siguientes:

- Líneas de ensamble
- Sistemas de manufactura flexible
- Producción por lotes
- Justo a tiempo (JAT) y Sistemas de producción KANBAN.
- Sistemas de colas. (Para servicios o manufactura tales como líneas de empaque).
- Optimización de la distribución en planta y el manejo de materiales.
- Servicio Financieros
- Logística
- Reingeniería de Negocios
- Evaluación, planeación y re-diseño de sistemas de servicios

ProModel es capaz de modelar aún los sistemas más complejos. Debido a que el ProModel provee un enfoque directo e intuitivo a la modelación, es atractivo a profesores de programas en ingeniería y administración quienes están interesados en enseñar los conceptos de modelación y análisis sin tener que enseñar programación.

Mientras que la mayoría de los sistemas pueden ser modelados al seleccionar un conjunto completo de elementos del ProModel (ejem. Recursos, tiempos muertos, locaciones, entidades, etc.) y modificar los parámetros apropiados, se provee también una capacidad completa de programación si es necesaria para modelar situaciones especiales. Funciones predefinidas incluyen funciones lógicas if-then-else, expresiones Booleanas, variables, atributos, arreglos y además acceso a hojas electrónicas y archivos de texto externos.

Para aquellos que prefieren códigos lógicos complejos usando un lenguaje de programación tal como C++ o Visual Basic, se pueden encadenar dinámicamente subrutinas externas a el modelo y llamarlas desde cualquier parte del modelo al momento de su ejecución. El ProModel también puede ser controlado como un objeto COM, y ejecutarse directamente desde aplicaciones externas tales como Microsoft Excel o PowerPoint, o desde una interfase VB. De esta forma, ProModel permite a todos los miembros del equipo de toma de decisiones usar las herramientas con las que se siente mas a gusto, proveyéndoles una flexibilidad total.

El ProModel también provee varias funciones de distribución predefinidas, las cuales junto con secuencias de números aleatorios, proporciona valores aleatorios de acuerdo a la distribución estadística. Para ayudar al usuario en la selección de la distribución de probabilidad apropiada para un conjunto de datos, el programa Stat:Fit es incluido en el ProModel. El Stat:Fit es un software de ajuste de datos que ajusta distribuciones analíticas a los datos del usuario.

El desarrollo del modelo es completamente gráfico y orientado a objetos. A su máxima extensión posible, todos los datos son dados gráficamente con información agrupada por tipo de objeto y representada en forma de tabla para un acceso rápido e intuitivo. Por ejemplo, cuando el modelador define una máquina el modelador también define el icono de la máquina, su capacidad, características de tiempos muertos, reglas de entrada y salida de datos, estadísticas deseadas, etc.

El ProModel compila con los estándares GUI (Graphical User Interface), el cual significa individuos familiarizados con otros programas de Windows tales como procesadores de datos y hojas electrónicas no tengan problema aprendiendo como usar el ProModel. Esta forma de suministrar los datos minimiza la curva de aprendizaje para principiantes y maximiza la eficiencia para modificar modelos grandes y complejos.

Una característica única en el ProModel es la habilidad de invocar menús tipo Popup, dependiendo del contexto actual, este facilita al usuario en definir cualquier estatuto o expresión. Este constructor lógico permite que pueda darse cualquier estatuto o expresión usando el mouse. Además elimina la necesidad de recordar el nombre de una variable o de otro elemento que el usuario desee hacer referencia, a través de seleccionar su nombre de una lista contenida en un recuadro.

Proporciona documentación conveniente en línea a través del sistema de ayuda integrada del ProModel y un tutorial en línea. El sistema de ayuda usa el sistema de ayuda del Windows que permite máxima flexibilidad para buscar cualquier cosa desde la sintaxis del comando hasta la descripción de la construcción del modelo. El ProModel también provee tutoriales que contienen lecciones rápidas sobre como construir el modelo, como ejecutarlos, como tener acceso a reportes, y como modelar varias aplicaciones con el Software.

Para reducciones futuras del tiempo de desarrollo, el ProModel provee capacidades de fusión de modelos que permiten a varios modelos individuales trabajar de forma separada en diferentes secciones de un modelo mayor. Adicionalmente, celdas que son comúnmente definidas o usadas en decisiones lógicas pueden ser almacenadas como plantillas de submodelos las cuales eliminan la necesidad de "re-inventar la rueda" con cada modelo. Estas plantillas pueden ser tener parámetros especialmente diseñados que pueden ser cambiados por el usuario.

El desarrollo de la animación es integrado con la definición del modelo. Un inconveniente de muchos softwares de simulación es que su desarrollo de animación depende del desarrollo del modelo de simulación. Esto hace que sea lento e inconveniente para los ingenieros al usar la animación como una herramienta de validación/verificación. ProModel integra el desarrollo de la definición sistema y animación en un proceso. Mientras que defina la ruta de las entidades por las locaciones, transportadores, rutas de VGA (Vehículos guiados automáticamente) y otros elementos, el usuario desarrolla esencialmente la distribución de forma animada.. La pantalla de la distribución puede ser cambiada en su escala para representar la distribución actual de la fábrica.

Los resultados de la simulación son informativos y pueden ser mostrados en forma tabular y gráfica. Muchos otros softwares de simulación requieren comandos especiales para generar estadísticas que son difíciles de interpretar para usuarios no familiarizados con la simulación. ProModel permite la selección rápida y conveniente de los reportes de todas las medidas de desempeño del sistema. Los reportes de resultados de varias corridas de simulación pueden ser comparadas en una sola gráfica.

EL ProModel se puede correr en cualquier computadora Pentium estándar o más rápida con sistema operativo Windows 95, Windows 98, Windows NT, Windows 2000, Windows Milenium,, o Windows XP. Las licencias están disponibles tanto para plataformas de un usuario como cómo para plataforma de redes. El ProModel no requiere ningún tarjeta gráfica especial o monitores especiales, haciendo conveniente y de costo efectivo para las compañías e instituciones académicas usando PC's estándar.

PROMODEL Corporation

*1875 S. State Street, Suite 3400
Orem, UT 84097
Phone: (801) 223-4600
Fax: (801) 226-6046
<http://www.promodel.com/>*

Teléfono 01 800 8 SIMULA ó 01 800 8 746852 Sin Costo

FAX 0181-8348-2270 y 01-800-112-7890 Fax Sin Costo

Dirección

Gonzalitos Norte 106-2 Monterrey NL 64620 MEXICO

<http://www.promodel.com.mx>

7.5.5 Simulador Taylor Ed.

El Taylor Ed es un paquete de software desarrollado por la compañía fabricante de software FLEXIM. El precursor del Taylor Ed fue el Taylor II, pero los desarrolladores hicieron la nueva versión partiendo de desperdicio. A pesar de esto, existen similitudes entre los dos paquetes, tales como la visión del usuario y la arquitectura del software. El precio del Taylor ED es la mitad del precio del WITNESS y cinco veces mas que el SIMUL8 y el EXTEND.

La integración del Taylor Ed con otro software es manejado por un encadenamiento DDE, el cual distribuye la información hacia y desde el EXCEL, por ejemplo, u otro software para el manejo de datos estructurados. Este DDE usa diferentes formatos de ases de datos, tales como SQL o XML. Esto es continuo las conexiones pueden tomar lugar antes o después que la simulación es realizada.

El concepto de modelación en el Taylor ED es el mismo que el desarrollador uso en el Taylor II. Este concepto es diferente de otros softwares. En el Taylor ED, todo es un átomo (producto, máquina, el modelo mismo, la aplicación del software), comparado con el SIMUL8, el cual consiste de cinco bloques de construcción, y el EXTEND el cual consiste de más de 100 bloques de construcción. El flujo de conexiones entre los átomos en Taylor ED esta determinado a través de canales, que es lo mismo que los encadenamientos en otros softwares.

El Taylor Ed sale del mercado dejando su lugar al simulador FLEXSIM. El futuro de la simulación de procesos es el FLEXSIM, que es un simulador orientado a objetos basado en un ambiente de Windows® para modelar flujos de eventos discretos como manufactura, manejo de materiales, y flujo en oficinas en una asombrosa realidad virtual de 3D.

Completamente orientado a objetos con una integración completa de C++ creada gráficamente usando la animación virtual ED excepcionalmente intuitiva de atrapa y suelta, una interfase fácil de aprender. La insuperable flexibilidad y poder del FLEXSIM es la perfecta herramienta para ayudar a los ingenieros, administradores, y tomadores de decisión a visualizar y probar operaciones propuestas, procesos y sistemas dinámicos en una realidad virtual de 3D.

Es indispensable para modelar modelos complejos que son susceptibles de fallar, ser interrumpidos y tener cuellos de botella. A través de la modelación de sistemas en avance de múltiples escenarios “que tal sí” pueden ser explorados sin interrumpirlos, costos o riesgos que esto significa en la vida real.

FLEXSIM usa una librería de objetos para modelar objetos de la vida real, procesos y sistemas. El software es escrito en C++, el actual lenguaje orientado a objetos mas usado. Todos los recursos para la construcción de modelos en FLEXSIM son objetos, ya sean productos, modelos, tablas, registros, librerías, la interfase gráfica del usuario (GUI), o la aplicación en si misma. Los objetos pueden heredar atributos y comportamientos de otros objetos, contener otros objetos, crear y destruir objetos, mover objetos dentro y fuera de ellos o auto destruirse.

Los objetos desarrollados para un modelo pueden rápidamente ser almacenados en librerías para ser usadas en otros modelos, reduciendo el consumo de tiempo, y la duplicidad de esfuerzo.

El FLEXSIM viene con una extensa librería de objetos robustos listos para usarse. Los usuarios pueden rápidamente modificarlos usando el editor de objetos integrado, o puede crear el propio partiendo de la nada usando el C++ o el poderoso Flexscript una librería precompilada del código C++ que puede controlar virtualmente cada aspecto del programa.

La característica de la arquitectura del FLEXSIM son su apertura e interconectividad. El FLEXSIM esta totalmente integrado con el C++, tal que los usuarios puedan rápidamente modificar el FLEXSIM para reunir las necesidades específicas sin tener que aprender el código apropiado.

Toda la animación es OpenGL y todas las gráficas son del estándar industrial de objetos 3DS, DXF, WRL, o STL. Los resultados pueden ser exportados vía DDE, ODBC, y conectores Windows. Otras aplicaciones complementarias como Expert Fit, OptQuesy, y VISIO™ están también compiladas para agregar flexibilidad y facilidad de uso.

El FLEXSIM se encadena a cualquier base de datos ODBC (como el Oracle o el Access), a estructuras de datos comunes (como archivos de texto, de Excel o Word), y virtualmente a cualquier dispositivo de hardware que pueda ser conectado a la PC.

Otros productos de FLEXSIM:

Flexsim GP, Flexsim WM, Flexsim Fabmodeler, Flexsim Port, y Flexsim SANS

Flexsim Software Products, Inc.

University Office Park

1366 South 740 East

Orem, Utah 84097

801.224.6914

801.224.6984

e-mail: info@flexsim.com

www.flexsim.com

7.5.6 Simulador WITNESS

EL software Witness fue desarrollado durante los 1970's por AT&T Istel y es ahora distribuido por el Grupo Lanner (establecido en 1996) del Reino Unido. El paquete usa ahora una base mas madura, lo cual probablemente sufre del dilema del innovador. El WITNESS fue previo, y posiblemente todavía es el líder en el mercado, aunque solamente en el número de los modelos producidos. El costo de la licencia del WITNESS es 10 veces más que el costo de compra del EXTEND o SIMUL8. El soporte es de alguna forma más costoso en el caso del WITNESS, aproximadamente tanto como una nueva licencia del paquete. Por otro lado WITNESS es uno de los más usados, mas confiables y mejor conocidos entre los softwares SED (Simulación de eventos discretos). El concepto de construcción de modelos en WITNESS consiste en la construcción de bloques similares a los de SIMUL8 excepto por los bloques Inicial y Terminal.

El WITNESS contiene muchos elementos para manufactura discreta de partes y es fuertemente orientado a máquinas. Por ejemplo, las máquinas pueden ser sencillas, en lotes, producción, ensamble, multi-estaciones, o multi-ciclo. Las bandas transportadoras pueden acumular o no acumular. Existen opciones para la mano de obra, vehículos, y cuadrillas de trabajo. El WITNESS

también contiene elementos para procesamiento continuo incluyendo flujo de fluidos a través de procesadores, tanques y pipas.

Se pueden especificar variables y atributos. Las partes que llegan pueden ser programadas usando un archivo. Se pueden usar funciones y distribuciones para especificar tiempos de operación y para otros propósitos. Los tiempos muertos de las máquinas pueden programarse sobre la base de operación, tiempo de uso, o tiempo disponible. La mano de obra es un recurso que puede ser preferenciado, usando un sistema de prioridades, y ser una base programada para las condiciones actuales del modelo.

La lógica de rastreo y conducción permite hacer requerimientos para distintos tiempos en los trabajos, acelerar y desacelerar vehículos, estacionarse cuando este ocioso, cambiar destinos dinámicamente. Muchos son posibles tipos de rutinas lógicas además del estándar jalar y empujar. Por ejemplo, se pueden especificar las condiciones If-then-else.

Se pueden emplear como acciones en la simulación constructos de programación, desarrollados al principio y final de la simulación de eventos, tales como for-next, while-end y la etiqueta go-to. Los C-LINKS permiten programación detallada y subrutinas que pueden agregarse al modelo del WITNESS. El usuario puede observar a un elemento en cualquier momento y determinar el estatus de la parte.

La depuración puede llevarse a cabo deteniendo el modelo, cambiando los parámetros deseados y continuando con el modelo desde el mismo punto del tiempo de simulación.

Una animación es construida junto con la definición del modelo. Esta retroalimentación animada y estadística puede ser activada o apagada durante la ejecución. Se pueden realizar muchos cambios al modelo en cualquier momento.

Capacidades de experimentación internas están disponibles desde la barra del menú. Los resultados de los experimentos son salidos a un archivo CSV por omisión, u otros tipos de archivos seleccionados por el usuario. El formato del archivo CSV permite que el paquete estadístico interno crear intervalos de confianza.

Lanner Group GmbH
Hansaallee 201
40549 Düsseldorf Germany
Tel: +49 (0) 211 530 63000
Fax: +49 (0) 211 530 63022

<http://www.lanner.com>

7.57 Simulador AIM

El AIM (Analyze for Improving Manufacturing) de la corporación Pritsker, es el componente de simulación de FACTOR (FACTOR/AIM). Otros componentes son el Administrador de Producción de FACTOR y FI-2. El Administrador de Producción de FACTOR desarrolla operaciones detalladas del planeación y programación de operaciones, promesa de órdenes, liberación de costos, y de la cadena suministro-abastecimiento. El FI-2 es un tablero de calendarización gráfica interactiva. Todas esas aplicaciones usan la misma base de datos.

El AIM esta disponible para la plataforma OS/2. LA actual versión del AIM es la 6.0. Los modelos son construidos gráficamente con iconos que representan máquinas, operadores, transportadores, y así sucesivamente, colocados directamente en la pantalla. Las animaciones son creadas en una ventana virtual. Durante la simulación, el modelo puede ser detenido para verificar su estatus o añadir otros componentes, entonces continuar con la simulación. Los datos son actualizados dinámicamente y mostrados mientras que la simulación se realiza. Una gráfica dinámica de Gantt se aporta para dar seguimiento al estatus de la máquina y el operador. Los niveles de inventario y la utilización de los materiales pueden también ser graficados dinámicamente. Los resultados incluyen gráfica de barras, gráficas de pastel, y gráficas del nivel de inventarios. Alternativamente, la información puede ser transferida a otro software para realizar gráficas de presentación.

600 W. Reichmuth Rd, Caja 338 del Po
Valle, Ne 68064 los EUA.
Contacto: Cuenta Fleissner
Teléfono: 866-aim-1553
E-mail: salesus@aim-online.com o fleissner@aim-online.com

Pritsker Corporation

8910 Purdue Road, Suite 600
Indianapolis, IN 46268-1170
Phone: (800) 428-7636
<http://www.pritsker.com>

7.5.7 Simulador Extend

Extend es un software orientado a apoyar el proceso de toma de decisiones, que permite visualizar el comportamiento y los resultados de un proceso en diversos escenarios definidos por el usuario, a un bajo costo y minimizando el riesgo de la implantación. La simulación de procesos permite evaluar comportamientos tanto en funcionamiento como en su etapa de diseño, sin incurrir en los costos de una implantación real. El realizar pruebas y modificaciones durante las etapas de diseño y planificación, permite ahorrar tiempo y dinero en las etapas posteriores de implantación y mantenimiento de los nuevos procesos.

Por la gran potencialidad que posee para la representación de sistemas complejos, la flexibilidad de su manejo y lo amigable de su interfaz, es una muy buena solución para empresas, u organizaciones en general, que deseen desarrollar modelos de simulación de sus procesos (de servicios, manufactura, negocios, administrativos, etc.) como parte de la evaluación y proyección de resultados de sus proyectos de transformación. Transformando sus flujos de proceso en modelos Extend y para efectuar un análisis a través de simulación, donde experimentará nuevas posibilidades, estudiará la respuesta a condiciones dinámicas y evaluará beneficios. Esto le permitirá alcanzar decisiones en forma más fácil y segura que métodos basados en intuición, permitiéndole instaurar cambios positivos para su operación. En particular, permite evaluar los supuestos que hay detrás de los modelos operacionales actuales, al analizar en forma sistémica el conjunto de variables que inciden en el proceso.

Con Extend se pueden modelar cambios organizacionales, probar escenarios, diseñar prototipos, analizar opciones de equipamiento, aplicar gestiones de mejoramiento continuo, incrementar productividad y calidad, y evaluar ideas antes de llevarlas a cabo. Permite que usted simule eventos discretos, continuos, y combinaciones de ellos. Virtualmente cualquier cosa que usted pueda imaginar puede ser fácilmente construida usando las bibliotecas de Extend o bloques prediseñados. No se requiere de programación, sin embargo, es posible si usted lo desea.

Los bloques de Extend están agrupados dentro de bibliotecas de acuerdo a su función; por ejemplo, los bloques más usados en la modelación de un evento discreto pueden encontrarlo en la biblioteca de Eventos Discretos (Discret Event Library). Tan sólo arrastre con el mouse los bloques que desea utilizar desde la biblioteca hacia la pantalla de construcción y listo. Conéctelos con el mouse, disponga de los parámetros apropiados en las cajas de diálogos, y ya está listo para correr su simulación.

Características del Extend:

- Permite ver los efectos de contar con eventos dependientes.
- Las fluctuaciones estadísticas, o variabilidad natural de los sistemas, son fácilmente modelables para dar realismo a los modelos.
- Modelamiento y simulación han sido complejas y caras para ser atractivas a ejecutivos; Extend ofrece una excelente relación costo / beneficio.
- El software permite modelar y simular cualquier tipo de sistema.
- Está orientado a manejo de objetos, lo que facilita el modelamiento a partir de sistemas reales de diversa complejidad.
- Gráfica y Animación incorporadas permite una mejor comprensión y visualización del modelo por terceras personas.
- Permite tratar diferentes procesos como sistemas relacionados, analizando el impacto de cambios en los mismos.
- Reportes de simulación entregan toda la información necesaria para tomar las mejores decisiones.
- Manejo jerárquico permite representar en forma más ordenada y clara los modelos.
- Recolección de estadísticas en forma gráfica complementa la información necesaria para evaluar los modelos.
- Parámetros simples para representar y medir la realidad de un modelo:
 - Tiempo por tarea o actividad
 - Tiempos de transferencia o traslado
 - Inventario en cada etapa del proceso
 - Calidad del resultado en cada etapa
 - Productividad de los recursos
- Tiempos de espera y flexibilidad

- Permite efectuar un estudio de tiempos para mejorar la productividad global y local de un sistema.
- Estudio de volúmenes de trabajo para detectar recursos con capacidad restrictiva

Imagine That, Inc.
6830 Via Del Oro, Suite 230
San Jose, CA 95119 USA

Email
extend@imaginethatinc.com.

Fax
1.408.629.1251
Phone
1.408.365.0305

<http://imaginethatinc.com/>

7.5.8 Simulador ARENA

El ARENA de la Systems Modeling Corporation, es un paquete de simulación y animación extendible. Se intenta proveer el poder del SIMAN para aquellos quienes aprender un lenguaje es una incomodidad, también como resaltar el uso de las herramientas usadas por los modeladores del SIMAN. Considere que una persona, diferente al analista de la simulación, desea usar el SIMAN. Actualmente, el o ella deben entender los bloques usados en el modelo y los elementos usados en el experimento para proceder. Usando las plantillas de solución para la aplicación del ARENA, el usuario puede extraer el modulo, colocarlo es su lugar apropiado, parametrizarlo sin aprender el lenguaje SIMAN. El lenguaje SIMAN para los modeladores, el ARENA intenta incrementar su funcionalidad, eliminando la necesidad de escribir códigos similares en diferentes modelos.

El SIMAN es la máquina del lenguaje y Cinema el sistema de animación sobre el cual se construye el ARENA. Otros productos incluidos en ARENA son un analizador de entadas y un analizador de resultados.

Con el ARENA, un modelo de simulación se construye seleccionando un módulos que contiene las características completas del proceso. Por ejemplo, un módulo de inspección puede modelar un proceso de inspección. El modulo se coloca en una ventana y una caja de diálogo aparece en la cual el usuario entra sus datos y elige opciones. Una vez que los módulos son colocados y las preguntas contestadas, el ARENA se ejecuta un modelo totalmente animado del proceso actual o del propuesto.

Los módulos pueden se organizados en plantillas especializadas para diferentes dominios de aplicación. Una vez que los modelos son creados, se transforman en paquetes de auto contenido lógicos que pueden re-usarse en otros modelos. Con esta habilidad para adecuarse, el ARENA puede ser usado para crear plantillas para una compañía, departamento o persona específica

usando un lenguaje y gráficos significativos que son apropiados para usuarios poco frecuentes de la simulación.

Los constructores del modelo tienen un control completo sobre la funcionalidad del software cuando se usa la edición profesional para construir módulos y plantillas. Un usuario final con la edición estándar puede construir modelos usando únicamente plantillas de alto nivel.

El término “módulos” es usado para representar la construcción de bloques disponibles para la creación de modelos. La característica más fundamental del ARENA es que un analista de simulación puede construir una definición del módulo para ser usados por otros en un medio ambiente de modelación jerárquica. Estas definiciones del módulo pueden ser combinadas para crear módulos. El SIMAN basa sus módulos al nivel más bajo posible del módulo. Estos corresponden al constructor básicos del SIMAN (bloques y elementos). Todos los otros módulos, llamados módulos derivados, son construidos desde los módulos base u otros módulos derivados. Una vez construidas, las plantillas incrementan la velocidad a la cual los módulos pueden ser construidos, y ayudan en la comprensión para aquellos que no están familiarizados con los bloques y elementos del SIMAN. Las plantillas proveen a los modeladores con un dominio específico AST, o una plantilla de aplicación de solución. Por ejemplo, actualmente esta disponible la plantilla de fabricación de cierre (WFT), la plantilla de reingeniería de procesos (BPR), y una plantilla de manufactura avanzada (AMT). Las plantillas pueden ser construidas por quienes comprenden la edición profesional.

El Cinema V basado en vectores está contenido en el ARENA. Esta capacidad de animación está integrada con los módulos del ARENA. Por ejemplo, cuando se añade un módulo para representar un proceso de manufactura, un modelador puede obtener ambos la modelación lógica para representar el proceso, así como los componentes del Cinema representando el trabajo en proceso, y el estatus del recurso (ocupado, ocioso, en reparación, etc.)

Las características más relevantes de la aplicación ROCKWELL ARENA son:

ARENA es un simulador de sistemas de eventos discretos.

- Utiliza el lenguaje de simulación SIMAN
- El código interno en SIMAN puede evaluarse, modificarse o adicionarse de subrutinas en lenguaje C, Fortran, etc.
- Permite programar visualmente mediante asociación de bloques (Crea modelos de simulación sin la necesidad de codificar programas)
- Admite simulación continua y discreta
- Gran flexibilidad de uso
- Permite la programación a bajo nivel
- Proporciona un entorno gráfico para visualizar la evolución de los sistemas simulados (Permite mostrar la animación del modelo construido)

Rockwell Software Inc.
Human Resources, Department E1
2424 S. 102nd St.
West Allis, WI 53227

<http://www.software.rockwell.com>

Systems Modeling Corporation

504 Beaver Street

Sewickley, PA 15143

Phone: (412) 741-3727

Fax: (412) 741-5635

<http://www.sm.com/>

7.5.9 Simulador SIMUL8

El SIMUL8 desarrollado por Visual Thinking Internacional Ltd en Escocia, Reino Unido en 1994, es frecuentemente promovido como un paquete para análisis rápidos y es frecuentemente usado en centros educativos.

SIMUL8 es un software para Simulación de Eventos Discretos. Permite al usuario crear un modelo visual del sistema que se está investigando, dibujando objetos que son sacados directamente sobre la pantalla. Los objetos típicos pueden ser colas o puntos de servicio. Las características de los objetos pueden definirse en términos de, por ejemplo, capacidad o velocidad.

Cuando el sistema ha sido modelado, se puede emprender la simulación. El flujo de trabajos en el sistema se muestra por animación en la pantalla tal que se puede evaluar que tan apropiado es el modelo.

Cuando la estructura del modelo se ha confirmado se pueden realizar numerosos ensayos que permitan describir el desempeño del sistema estadísticamente. Los estadísticos de interés pueden ser tiempo promedio de espera., utilización de recursos, etc.

Las opciones de interfase de SIMUL8 (que determina la terminología y los iconos usados) se presentan en tres diferentes ambientes: servicio de salud, servicios de industria, fabricación. Este manual brinda al lector bases para el uso de SIMUL8 con base en el ambiente de fabricación.

El concepto de construcción del modelo en SIMUL8 es muy simple de entender y fácil de aprender como usarlo. Existen en la actualidad solo cinco artículos que deben ser entendidos, y todos ellos trabajan en una forma similar. Estos cinco artículos son:

- Punto de entrada de trabajos (fuente)
- Centro de trabajo (ejem: estación, servidor, máquina, etc.)
- Área de almacenamiento (ejem: cola enfrente de una estación, etc.)
- Salida de trabajos
- Recursos (Mano de obra, herramientas, etc.)

Por un pequeño pago extra el usuario puede obtener varias adiciones, incluyendo análisis de costos, optimizador del modelo, y gráficas Gantt para reportes. Estas adiciones son fáciles de entender y son integradas en una interfase estándar.

El SIMUL8 es también compatible con VISIO, un botón del SIMUL8 aparece en el menú principal en VISIO si el SIMUL8 y VISIO son instalados en la misma computadora. Dando clic en este botón transforma un diagrama de flujo TQM en un modelo de SIMUL8 (un almacén se transforma en una unidad de almacenamiento en SIMUL8; Operaciones, procedimientos o transporte son transformados en centros de trabajo).

141 St James' Road
Glasgow
UK
G4 0LT

Tel: 44 141 55 26 888

Fax: 44 141 303 8388

Visual Thinking

55A Part Street East

Mississauga, Ontario L56 4PG

Canada

Email: info@visualt.com

7.5.11 Relación de vendedores de Software de simulación de eventos discretos

Software	Compañía	Teléfono	Fax	Clase	Precio
ARENA	Systems Modeling Corp. 504 Beaver St. Sewickley, PA	(412) 741-	(412)	P	na

	15143	3727	741-5635		
@RISK	Palisade Corp. 31 Decker Rd. Newfield, NY 14867	(800) 432-7457	(607) 277-8001	A (Excel)	\$395
AutoMod	AutoSimulations 655 Medical Drive Bountiful, UT 84010	(801) 298-1398	(801) 298-8186	P	\$18,000
AutoSched	AutoSimulations 655 Medical Drive Bountiful, UT 84010	(801) 298-1398	(801) 298-8186	S (sched)	\$41,000
Best Network	Best Consultants 21450 Chagall Rd. Topanga, CA 90290	(818) 340-1146	(818) 346-7098	S (netw)	\$1,650
COMNET III	CACI Products Co. 3333 N. Torrey Pines Ct. La Jolla, CA 92037	(619) 457-9681	(619) 457-1184	S (netw)	\$1,500
CSIM17	Mesquite Software 3925 West Braker Lane Austin, TX 78759-5321	(512) 305-0080	(512) 305-0009	P	\$450
Extend	Imagine That, Inc. 6830 Via Del Oro, Ste. 230 San Jose, CA 95119	(408) 365-0305	(408) 629-1251	P	\$695
Extend+BPR	Imagine That, Inc. 6830 Via Del Oro, Ste. 230 San Jose, CA 95119	(408) 365-0305	(408) 629-1251	S (wflo)	\$990
Extend+Manufacturing	Imagine That, Inc. 6830 Via Del Oro, Ste. 230 San Jose, CA 95119	(408) 365-0305	(408) 629-1251	S (mfg.)	\$990
FACTOR/AIM	Pritsker Corp. 8910 Purdue Rd., Ste. 500 Indianapolis, IN 46268-1170	(800) 428-7636	(317) 471-6525	S (mfg.)	na
GPSS/H	Solverine Software Corp. 7617 Little River Turnpike Suite 900 Annandale, VA 22003-2603	(800) 456-5671	(703) 642-9634	P	\$2,500
GPSS/PC	Minuteman Software P.O.	(800) 223-	(508)	P	\$1,995

	Box 171 Stow, MA 01775	1430	897-7562		
GPSS World	Minuteman Software P.O. Box 171 Stow, MA 01775	(800) 223-1430	(508) 897-7562	P	\$4,500
GSS	Prediction Systems, Inc. 309 Morris Ave. Spring Lake, NJ 07762	(908) 449-6800	(908) 449-0897	P	na
Hocus Simulation 8000	P-E International P.O.Box 840 Purcellville, VA 22132-0840	(703) 338-2234	(703) 338-2322	S	\$9,000
Maintsim	Harrison and Co. 8909 Glenbrook Rd. Fairfax, VA 22031	(703) 280-2202	(703) 280-2202	S (maint)	\$295
ManSim/X	TYECIN Systems, Inc. Four Main St. Los Altos, CA 94022	(415) 949-8501	(415) 949-8505	S (schd)	\$49,500
MAST Simulation Environment	CMS Research Inc. 627 Bayshore Drive Oshkosh, WI 54901	(414) 235-3356	(414) 235-3816	S (mfg.)	\$2,950
MEModel	PROMODEL Corp. 1875 S. State St. Suite 3400 Orem, UT 84058	(801) 223-4600	(801) 226-6046	S (hcre)	na
micro-GPSS	Ingolf Stahl, SSE Box 6501 S-113 83 Stockholm SWEDEN	+46 8 736 94 25	+46 8 30 47 62	P	\$700
Micro Saint with Action View	Micro Analysis & Design 4900 Pearl E Circle Drive Suite 201E Boulder, CO 80301	(303) 442-6947	(303) 442-8247	P	\$5,995
MODSIM II	CACI Products Co. 3333 N. Torrey Pines Ct. La Jolla, CA 92037	(619) 457-9681	(619) 457-1184	P	\$1,500
MOGUL	High Performance Software Inc. P.O. Box 292466 Dayton, OH 45429	(513) 438-5558		S (comp)	\$4,995

NETWORK II.5	CACI Products Co. 3333 N. Torrey Pines Ct. La Jolla, CA 92037	(619) 457-9681	(619) 457-1184	S (netw)	\$1,500
OPNET Modeler	MIL 3, Inc. 3400 Intr'l Dr., N.W. Washington, DC 20008	(202) 364-8390	(202) 364-6182	S (netw)	\$16,000
PASION	Stanislaw Raczynski P.O. Box 22-783 14000 Mexico D.F., Mexico	(525) 563-8543		L (Pascal)	\$250
PC Model for Windows	SimSoft, Inc. 1020 Fifield St. Brookings, OR 97415	(800) 469-0984	(503) 469-0327	S (mfg.)	\$3,000
ProModel	PROMODEL Corp. 1875 S. State St. Suite 3400 Orem, UT 84058	(801) 223-4600	(801) 226-6046	S (mfg.)	\$13,900
PROVISA	AT & T ISTEEL 25800 Science Park Dr. Cleveland, OH 44122	(216) 292-2668	(216) 292-2861	S (schd)	na
QueGAUSS	Aptech Systems Inc. 23804 S.E. Kent-Kangley Maple Valley, WA 98038	(206) 432-7855	(206) 432-7832	P	\$275
QUEST	Deneb Robotics, Inc. P.O. Box 214687 Auburn Hills, MI 48321	(810) 377-6900	(810) 377-8125	P	na
ServiceModel	PROMODEL Corp. 1875 S. State St. Suite 3400 Orem, UT 84058	(801) 223-4600	(801) 226-6046	S (busn)	\$13,900
SIGMA for Windows	Boyd & Fraser Pub. Co. One Corporate Place Ferncroft Village Danvers, MA 01923	(800) 225-3782	(508) 777-9068	P	\$4,995
SIMFACTORY II.5	CACI Products Co. 3333 N. Torrey Pines Ct. La Jolla, CA 92037	(619) 457-9681	(619) 457-1184	S (fact)	\$1,500
SIMNET II	SimTec Inc. P.O. Box 3492 Fayetteville, AR 72702	(501) 575-3158	(501) 575-7446	P	na

SIMPLE++	AESOP GmbH Koenigstrasse 82 70173 Stuttgart GERMANY	+49-711- 163590	+49-711- 1635999	L (C++)	Less than \$6,000
SIMPROCESS III	CACI Products Co. 3333 N. Torrey Pines Ct. La Jolla, CA 92037	(619) 457- 9681	(619) 457-1184	S (busn)	\$1,500
SIMSCRIPT II.5	CACI Products Co. 3333 N. Torrey Pines Ct. La Jolla, CA 92037	(619) 457- 9681	(619) 457-1184	P	\$1,500
SIMUL8	Visual Thinking Int'l 3 Robert Speck Parkway Suite 900 Mississauga, Ontario CANADA L4Z 2GS	(905) 279- 6154		P	\$395
SLAMSYSTEM	Pritsker Corp. 8910 Purdue Rd., Ste. 500 Indianapolis, IN 46268-1170	(800) 428- 7636	(317) 471-6525	P	na
SSS	M. A. Pollatschek, Management Technion Haifa 32000 ISRAEL	+972 4 829 4430	+972 4 823 5194	L (C)	\$0
Taylor II	F & H Simulations, Inc. P.O. Box 658 Orem, UT 84059- 0658	(801) 224- 6914	(801) 224-6984	P	na
Witness	AT & T Istel 25800 Science Park Dr. Cleveland, OH 44122	(216) 292- 2668	(216) 292-2861	P	na
WorkFlow Analyzer	Meta Software Corp. 125 Cambridge Park Dr. Cambridge, MA 02140	(617) 576- 6920	(617) 661-2008	S (busn)	\$10,000

Abreviaciones

A	Adición (add-on) para una hoja electrónica
busn	Aplicación a negocios

comp	Aplicación a computo
fact	Aplicación a fábricas
hcre	Aplicación a salud
L	Librerías de simulación para lenguajes de propósito general
na	No disponible, no conocido
netw	Aplicación a redes
maint	Aplicación a mantenimiento
mfg	Aplicación a manufactura
P	Lenguaje de programación para simulación
S	Sistema para una aplicación particular
schd	Aplicación a la programación y calendarización
wflo	Aplicación a flujos de trabajos

Fuente: <http://iew3.technion.ac.il/~moshep/simsurv.html>

Referencias Bibliográficas

- A. Law y W. Kelton, *Simulation Models and Analysis*, segunda edición, McGraw-Hill (1991).
- A. Alan Pritsker, *Introduction to Simulation and SLAM II* (3rd ed.), John Wiley & Sons, Inc., New York, NY, 1986
- A. M. Law and D. W. Kelton. *Simulation Modeling and Analysis*. Industrial Engineering and Management Science. McGraw-Hill Inc., 2nd edition, 1991.
- Banks, J., Carson, J.S., II, and Goldsman, D., "Discrete-Event Computer Simulation," *Handbook of Statistical Methods for Engineers and Physical Scientists*, 2nd ed., (H.M. Wadsworth, Ed.), McGraw-Hill, New York, 1998.
- Banks, J. [1994], "Software for simulation," en 1994 *winter Simulation Conference Proceedings*, ed. J.D. Tew, S.Manivannan, D.A. Sadowski, A.F. Seila, Association for computing Machinery, New York, NY, pag. 26-33

Banks, J., B. Burnette, J.D. Rose, and H. Kozloski. Forthcoming. SIMAN V and CINEMA 11. New York: John Wiley.

B. Concebís, "Discrete Systems Simulation", McGraw-Hill, 1994

Camm, J. and J. R. Evans. (1996). *Management Science: Modeling, Analysis and Interpretation*, South-Western, Cincinnati, Ohio.

C. Dennis Pegden, Randall P. Sadowski, Robert E. Shannon, Introduction to Simulation Using SIMAN, McGraw-Hill, Inc., New York, NY, 1995

C. Dennis Pegden, Deborah A. Davis, Arena: a SIMAN/Cinema-based hierarchical modeling system, Proceedings of the 24th conference on Winter simulation, p.390-399, December 13-16, 1992, Arlington, Virginia, United States

Charles R. Harrell, Jeffrey J. Leavy, ProModel tutorial, Proceedings of the 25th conference on Winter simulation, p.184-189, December 12-15, 1993, Los Angeles, California, United States

Ch. Harrel, B. Ghosh, y R. Borden, "Simulation using ProModel", McGraw-Hill, 2003 Second edition

Christos Alexopoulos, Andrew F. Seila, "Advanced Methods for Simulation Output Analysis", *Proceedings of the 1998 Winter Simulation Conference*, D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds

D. Matko, R. Karba, B. Zupancic, Simulation and Modelling of Continuous Systems, Prentice-Hall (1992).

David M. Profozich, David T. Sturrock, Introduction to SIMAN/Cinema, Proceedings of the 26th conference on Winter simulation, p.427-430, December 11-14, 1994, Orlando, Florida, United States

Evans, J. R. and D. L. Olsen. (1998). *Introduction to Simulation and Risk Analysis*, Prentice-Hall, Upper Saddle River, NJ.

Evans, J.R. (2000), "Spreadsheets as a Tool for Teaching Simulation," *INFORMS Transactions on Education*, Vol. 1, No. 1,

Goble, J. Simulation Conference, 1991. "Introduction to SIMFACTORYII.5," en 1991 *winter Simulation Conference Proceedings*, Winter Balci, O., Sadowski, R.P., Nance, R.E. Association for computing Machinery, New York, NY, pag. 136-139

Harrel Ch., Price, R [2003], "Simulation Modeling Using Promodel Technology," en 2003 *winter Simulation Conference Proceedings*, ed. Chick, S, Sánchez P.J. y Morrice D.J. Addison-Wesley (1989).

Hartvigsen, David. (2001). *SimQuick: Process Modeling with Excel*, Prentice-Hall, Upper Saddle

Hillier, F., Hillier, M y Lieberman, G, "Métodos Cuantitativos para la Administración", Ed. McGraw-Hill, 2002

J. Banks, J. S. Carson, and B. L. Nelson. *Discrete-event system simulation*.

Kelton, D.W., Sadowski, R., "SIMULATION WITH ARENA", Ed. McGraw-Hill, Nueva York 2002, 2ª edición.