



# SOFTWARE DE SIMULACIÓN

Ing. Eduardo Raffo Lecca

## RESUMEN

El artículo muestra la clasificación de software de simulación existente, y como estos pueden ser utilizados en los sistemas de manufactura.

## ABSTRACT

This articles summarizes the classification of existing software for simulation, and how they can be used in manufacturing systems.

## Introducción

Muchas propiedades en programación de modelos de simulación discreta, tales como:

- Generadores de números aleatorios.
- Generadores de variables aleatorias.
- Rutinas del siguiente evento.
- Avance de tiempo.
- Recopilación de estadísticas.
- Reportes, etc.

Han sido desarrolladas en lenguajes especiales, orientados a simulación, dejando la ardua labor de programación en FORTRAN, C, o PASCAL a lenguajes de simulación, los que incluyen facilidades de animación. Actualmente, existen cerca de 100 software de simulación, disponibles en una variedad de computadores (ver fig. 1).

## Lenguajes de Simulación y Lenguajes de Propósitos Generales

La importancia de escribir modelos de simulación en lenguajes de propósitos generales como FORTRAN, radica en:

- Permite conocer los detalles íntimos de la simulación.
- Es imprescindible, cuando no se dispone de software de simulación.
- Algunos modelos en lenguajes de simulación permiten interfaces con lenguajes generales específicamente FORTRAN (ocurre con SLAM II, SIMAN, GPSS).

Por otra parte, los lenguajes de simulación ofrecen mayores ventajas, porque:

- Automáticamente proveen muchas de las facilidades necesarias en la simulación del modelo.
- Proveen un natural ambiente para modelamiento de la simulación.
- Son fáciles de usar.
- Proveen una gran interacción entre edición, depuración y ejecución. Alcanzando algunos de ellos implantación de la ingeniería de software.

## Clasificación de los Software para Simulación.

Existen en el mercado dos grandes clases de software para simulación: los lenguajes y los simuladores. Un lenguaje de simulación, es un software de simulación de naturaleza general; y posee algunas características especiales para ciertas aplicaciones; tal como ocurre con SLAM II y SIMAN con sus módulos de manufactura. El modelo es desarrollado usando las instrucciones adecuadas del lenguaje y permitiendo al analista un gran control para cualquier clase de sistema.

Un simulador (o de propósitos especiales), es un paquete de computadoras que permite realizar la simulación para un ambiente específico, no requi-

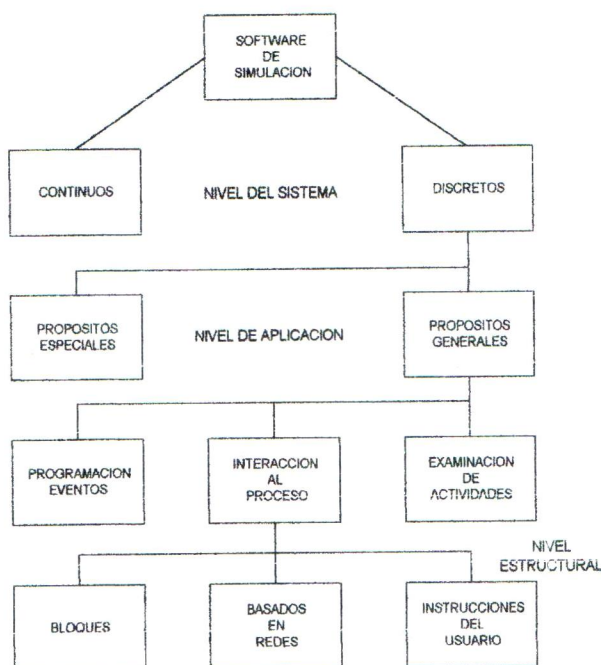


Fig. 1. Clasificación de los Software



riendo esfuerzo en programación. Hoy en día existen simuladores para ambientes de manufactura y sistemas de comunicación permitiendo un menor tiempo en el desarrollo del modelo, así como también contar con el personal sin experiencia en simulación.

Los simuladores son actualmente muy utilizados para análisis en alto nivel, requiriéndose únicamente agregar detalles en un cierto nivel, puesto que lo demás es estándar.

CACI Products Company autor de SIMSCRIPT II.5 es también autor de los simuladores SIMFACTORY II.5, NETWORK II.5 y COMNET II.5, muy utilizados en estos últimos tiempos para simulaciones de sistemas de manufacturas, redes de computadoras y redes de telecomunicaciones.

Para procesar transacciones en espera de un ordenamiento, un lenguaje de simulación debe proporcionar un medio automático de almacenamiento y recuperación de estas entidades. Atendiendo a la orientación del modelamiento de una simulación discreta, existen tres formas:

1. Programación de eventos.
2. Procesos.
3. Examinación de actividades.

Una programación al evento es modelada, identificando las características del evento y luego se escriben un juego de rutinas para los eventos con la finalidad de describir detalladamente los cambios que ocurren en el tiempo en cada evento. Lenguajes como SIMSCRIPT II.5 y SLAM II están orientados al evento.

Una interacción al proceso, es una secuencia de tiempos interrelacionados, describiendo la experiencia de una entidad a través del sistema. Por ejemplo en un modelo de colas, esta "historia" se traduce en el paso del tiempo del ingreso a la cola, ingreso al servidor, paso del tiempo en el servicio y fin del servicio (ver fig. 2). GPSS, SIMAN y SIMNET son orientados al proceso.

En el examen de actividades, el modelador define las condiciones necesarias al empezar y finalizar cada actividad en el sistema. El tiempo es avanzado en iguales incrementos de tiempo y en cada incremento de tiempo, las condiciones son evaluadas para determinar si alguna actividad puede estar empezando o terminando. El ESCL, es un lenguaje de simulación muy popular en Europa y fue desarrollado en FORTRAN. (Ver Tabla 1).

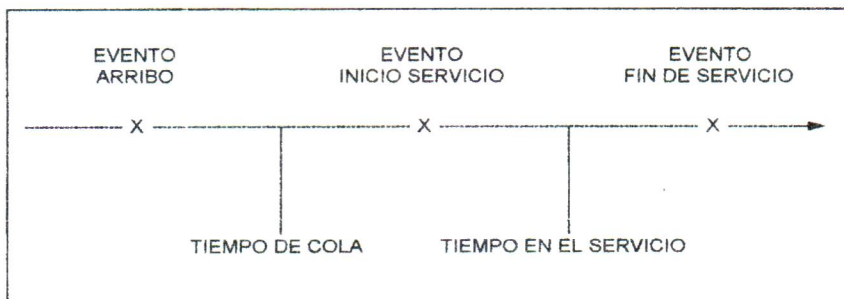


Fig. 2 Descripción de un proceso

### GASP IV

Es una colección de subrutinas FORTRAN, diseñadas para facilitar la simulación de secuencia de eventos. Cerca de 30 subrutinas y funciones que proveen numerosas facilidades, incluyendo:

- Rutinas de avance del tiempo.
- Gestión de listas de eventos futuros.
- Adición y remoción de entidades.
- Colección de estadísticas.
- Generadores de variables aleatorias.
- Reporte estándar.

El programador únicamente provee un *program main*, una rutina de actualización, rutinas de eventos, generadores de reportes personalizados y una subrutina denominada EVNTS. El programa *main* debe incluir la sentencia CALL GASP; siendo GASP una subrutina que determina el eminente evento, invocando a EVNTS; escrita por el usuario y obtiene el índice NEXT.

SOFTWARE	CONTINUO DISCRETO	PROPÓSITO GENERAL/ ESPECIAL	VERSIÓN PC/ MANIFRAME
Auto Mod	D	S	M
FACTOR	D	S	PC,M
GPSS V	D	G	M
GPSSH	D	G	M
GPSS PC	D	G	PC
MAP 1	D	S	M
PAC Model	D	G	PC
RESQ	D	G	M
SEE WHY	D	G	PC,M
SIMAN	D.C	G	PC,M
SIMFACTORY	D	S	PC
SIMPLE 1	D.C	G	PC
SIMSCRIPT	D.C	G	PC,M
SLAM II	D.C	G	PC,M
XCELL	D	S	PC

Tabla 1: Información básica en Software de Simulación.

GASP IV, es un lenguaje de simulación desarrollado por Alan B. Priestker y N. Hurst en 1973. es un lenguaje híbrido porque puede ser usado para programadores de simulación discretos, continuos y combinados; siendo el primero en integrar completamente estos dos ambientes de función del tiem-



po, GASP IV es un derivado del GASP II, y se diferencia por la definición del evento espacio-estado (*state space event*).

### SIMSCRIPT II.5

Desarrollado en la RAND Corporation por H. Markowitz en los inicios de los sesenta. SIMSCRIPT II.5, es un lenguaje de simulación con orientación al evento y al proceso; es híbrido porque posee facilidades para simulación de sistemas discretos y continuos. Un programador SIMSCRIPT II.5 consiste de las siguientes partes:

- Preamble
- Main program
- Rutinas de eventos.
- Rutinas ordinarias.

SIMSCRIPT II.5, producido por CACI Products Company (La Jolla, California); fue utilizado en el pasado en grandes y complejas simulaciones; como es el caso de los **modelos no orientados a colas**; por ejemplo modelos de combates militares. Se encuentra disponible en versión PC; destacando su ambiente de SIMGRAPHICS.

SIMSCRIPT II.5, está basado en **entidades, atributos y conjuntos**. Visualiza el mundo a ser simulado como un conjunto de entidades, que pueden ser descritas a través de sus atributos y los eventos que aparecen en el tiempo.

### SIMAN/Cinema

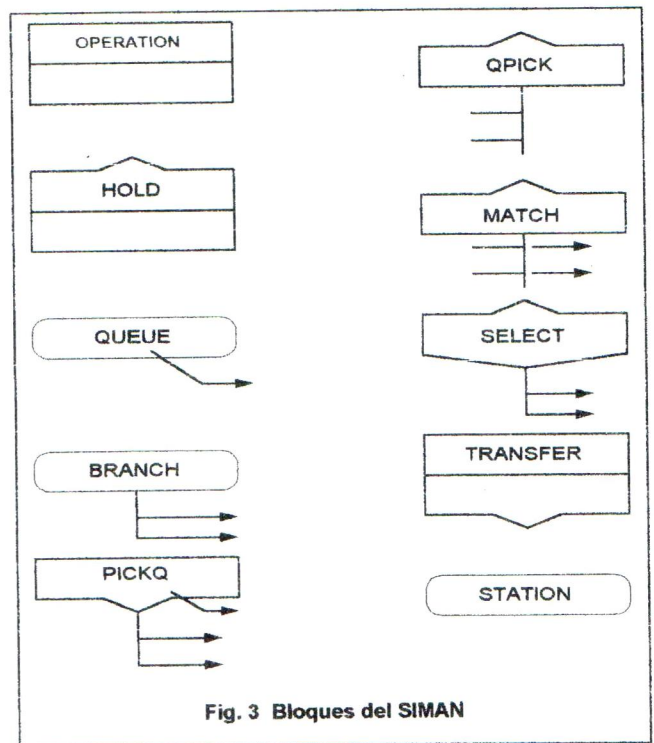
La versión original del SIMAN (Simulation and Analysis) fue desarrollada por Dennis Pegden, en la Universidad de Alabama, cuando era líder del grupo de desarrollo de la versión original de SLAM (basada en los software de GASP y Q-GERT de Pristker and Associates). Más tarde, Pegden inicia su trabajo en el Pennsylvania State University, donde lo diseña como un lenguaje de modelamiento para propósitos generales, incluyendo facilidades de manufactura, muy útiles en modelamiento de sistemas complejos de manufactura.

Desde su implementación inicial en 1984, ha sido continuamente refinado por System Modeling Corporation; y en 1998 y 1989, el lenguaje fue completamente rediseñado dando origen a SIMAN/Cinema.

El ambiente de modelamiento en SIMAN, se desarrolla entre el Modeling y el Experiment; en el primero se describe las componentes del sistema y sus interacciones, y en el segundo se definen las condiciones del experimento (longitud de la corrida, condiciones iniciales).

SIMAN, modela un sistema discreto usando la orientación al proceso; es decir, en un modelo de sistema particular, se estudian las entidades que se mueven a través del sistema. Una entidad para SIMAN es un cliente, un objeto que se mueve en la simulación y que posee características únicas, conocidas como atributos. Los procesos denotan la secuencia de operaciones o actividades a través del que se mueven las entidades; siendo modeladas por el diagrama de bloques.

Ud. construye un diagrama de bloque en un flow-chart gráfico, seleccionando y combinando bloques. Después interactivamente usando un editor especial se activa el generador automático de las sentencias del modelo, desde el ambiente gráfico. Los bloques de SIMAN, se clasifican en 10 tipos básicos. (ver fig. 3).



### SLAM II

El SIMSCRIPT y el GASP IV son los lenguajes de programación de eventos más destacados.

SLAM es un descendiente de GASP IV, que ofrece también recursos de simulación de redes y continuos, estando ambos codificados en FORTRAN.

Desde los lenguajes orientados a los procesos, existen representación de modelos, en bloques como GPSS y SIMAN y los basados en redes como Q-GERT y SLAM.



Con la llegada del PERT, se plantearon situaciones de redes complejas, en tanto a ramificación por efecto de una decisión y loop para conseguir que varias actividades se realicen de modo repetitivo; trayendo consigo el desarrollo del GERT (Graphical Evaluation and Review Technique), por Pritoker y Elaghraby; quienes lo aplicaron para el programa Apolo.

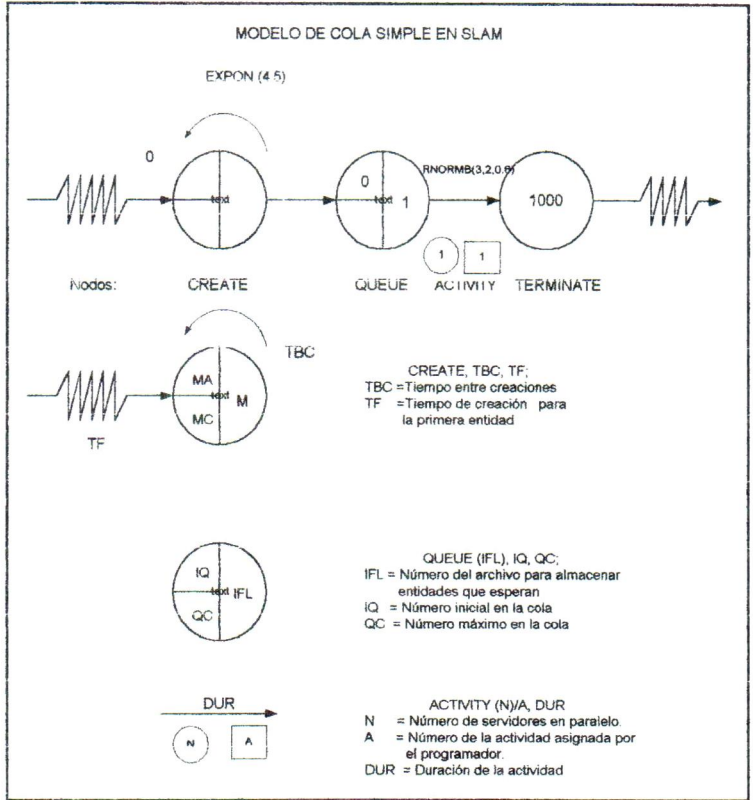
El lenguaje Q-GERT, significó la respuesta al cálculo de estimación de probabilidades de terminación en cada nodo y la distribución de tiempos y costos para la realización de cualquier nodo. La estructura básica de un modelo de simulación Q-GERT es una red compuesta de nodos y actividades (bifurcaciones). SLAM es una variante de Q-GERT, que ofrece recursos de eventos de redes y discretos (y también simulación continua).

SLAM II (Simulation Language for Alternative Modeling) es un lenguaje de simulación por el cual se pueden construir modelos con orientación al proceso o al evento. SLAM fue desarrollado en 1979 por Dennis Pedge y Alan Pritsker y es distribuido por Pritsker Corporation (Indianapolis, Indiana). La parte de SLAM que se orienta a los procesos, emplea una estructura reticular compuesta por símbolos de nodos y ramas; tales como colas, servidores y puntos de decisión. Modelamiento significa incorporar esos símbolos a un modelo de red, que representa el sistema; y en donde las entidades (items) pasan a través de la red. SLAM contiene un procesador que convierte la representación visual del sistema a un conjunto de sentencias.

La parte orientada a los eventos, permite incluir rutinas en FORTRAN para las relaciones lógicas y matemáticas que describen los cambios en los eventos.

Un modelo continuo, es especificado por las ecuaciones diferenciales o de diferencia, el que describe la conducta dinámica de las variables de estado. El modelador codifica esas ecuaciones en FORTRAN, empleando un juego especial de arreglos de almacén SLAM.

El SLAM, simplifica el modelamiento de sistemas complejos, combinando el uso fácil de lenguaje de proceso como GPSS y Q-GERT, con la potencia y flexibilidad del lenguaje de eventos GASP IV.



"Los lenguajes de simulación facilitan grandemente el desarrollo y la ejecución de simulaciones de sistemas complejos del mundo real". (Banks & Carson)

Cada lenguaje posee una orientación a situaciones del mundo real o world pudiendo ser:

- Programación de eventos.
- Orientado al proceso.
- Examinación de actividades

Generalmente el modelo resultante en estos lenguajes, será con la orientación al proceso o el evento; o una combinación de ellos.

FORTRAN, es un lenguaje de programación científica, no diseñado para usarse en simulación. El analista cuando recurre a FORTRAN, lo más probable es que utilice la orientación del "secuenciador de eventos" (events-scheduling).

GPSS, es un lenguaje de simulación orientado al proceso, especialmente a las transacciones (un caso especial de la orientación al proceso). Diseñado para la simulación de sistemas de colas.

FORTRAN y GPSS, son muy utilizados para modelamiento de simulación de sistemas discretos.

SIMSCRIPT y SLAM, son lenguajes de programación para simulación de alto nivel; y permiten facilidades en el diseño de la construcción del modelo. SLAM, es un sistema basado en FORTRAN y contiene al subconjunto GASP. GASP es un conjunto de subrutinas escritas en FORTRAN para facilitar simulaciones orientadas al evento.

SIMSCRIPT, contiene como subconjunto, a un complejo lenguaje de programación científica comparable con FORTRAN, PL/1 y ALGOL.

GPSS junto con SIMSCRIPT, han resultado ser los lenguajes de simulación más ampliamente usados

Tabla 3: Resumen