

MODELO COMPUTACIONAL DE SIMULACIÓN

DE PROCESOS CONSTRUCTIVOS

“MOCSPROC”

Juan Camilo Granados Riveros

RESUMEN: El desarrollo de herramientas especializadas en simulación de procesos constructivos es una operación compleja que requiere alto grado de investigación. Actualmente no se tiene acceso al cuerpo de programación de herramientas reconocidas dentro de este medio, razón por la cual existe la necesidad de crear un código propio y flexible que se adapte a las condiciones. MOCSPROC se consolida como el primer paso en el desarrollo de una herramienta de simulación de procesos constructivos orientada por eventos, estructurada por módulos y programada en Visual Basic 6.0 con interfaz Microsoft Excel. Su misión es obtener información de la ejecución y comportamiento de procesos constructivos, bajo condiciones y restricciones específicas, y generar reportes sobre estos resultados. La información que obtiene MOCSPROC es útil para toma de decisiones en las etapas de planeación y diseño de procesos constructivos, cuyo objeto es el mejoramiento de la productividad de los mismos.

PALABRAS CLAVES

Simulación, modelación, procesos constructivos, productividad, simulación discreta, algoritmo de simulación y simulación Monte Carlo.

INTRODUCCIÓN

El desarrollo de nuevas herramientas de simulación para la industria de la construcción es actualmente un fin gerencial primordial para representar de una manera muy aproximada lo ocurrido en la realidad y tenerlo en cuenta para tomar decisiones acertadas, previas a la ejecución de los procesos constructivos. Aunque su incursión en este mundo no ha sido fácil por la dificultad de plantear modelos que se adapten al entorno; por el tiempo de investigación tan alto que es necesario emplear y por las características agrestes del medio; hoy por hoy se cuenta con herramientas de simulación en computador que facilitan el

análisis y la optimización de procesos relacionados con la construcción. (Martínez 1996)

En un proceso de toma de decisiones no es fácil llegar a una solución óptima utilizando métodos de decisión complejos como modelos matemáticos y métodos que demandan gran cantidad de tiempo y costo. Por esta razón, la simulación es un método realista que se ajusta a las condiciones del medio, es flexible, se realiza en el tiempo disponible y no necesita inversiones muy elevadas.

La simulación en computador se define como el proceso de diseño de un modelo lógico-matemático de un sistema del mundo real y la aplicación del modelo en el computador (Pristker 1979), para conocer información que no se posee por dificultades económicas, de tiempo o de técnicas necesarias para su obtención. Entonces, los modelos de simulación pueden ser considerados como una versión de laboratorio de un sistema, para realizar experimentos que pueden ser mejorados paulatinamente.

En síntesis, el propósito de la simulación es obtener información para complementar la investigación, planeación y diseño de un sistema o proceso y así evitar errores que puedan ser costosos en el futuro. De esta manera se disminuye la incertidumbre del comportamiento del sistema y se tiene la oportunidad de tomar decisiones acertadas que seguramente aumentarán la productividad de los procesos.

Los cambios en los estados de un sistema pueden ocurrir continuamente sobre el tiempo o en instantes discretos en el tiempo. Estos términos describen al modelo y no al sistema real. De hecho, es posible modelar el mismo sistema ya sea con cambios discretos o cambios continuos. En la mayoría de simulaciones el tiempo es la variable independiente, mientras las demás variables están en función del tiempo y son las variables dependientes.

~~La simulación discreta~~ ocurre cuando las variables dependientes cambian discretamente en puntos específicos del tiempo de simulación conocidos como eventos, mientras que en ~~la simulación continua~~ las variables dependientes del modelo cambian continuamente en cualquier momento del tiempo de simulación. La mayoría de procesos constructivos pueden ser efectivamente modelados utilizando la simulación discreta por eventos. (Martínez 1996)

Las herramientas específicas de simulación pueden ser clasificadas como simuladores o como lenguajes de simulación (Law and Kelton 1991). Los simuladores son paquetes de computador que permiten la simulación de una clase específica de sistemas con poco o nada de programación, mientras los lenguajes de simulación son muy generales, aunque pueden caracterizarse para modelar cierto tipo de aplicaciones. En general, los lenguajes de simulación tienen la habilidad de modelar casi cualquier tipo de sistema. (Martínez 1996)

Los simuladores y lenguajes de simulación pueden adoptar uno de los diferentes métodos o estrategias. Existen tres estrategias que son generalmente reconocidas: Orientación por Eventos (ES) es gobernada por un calendario y ejecución de subrutinas (eventos) que como consecuencia programa la ejecución de otras subrutinas; Orientación de Seguimiento de Actividades (AS) se enfoca en identificar las actividades y las condiciones bajo las cuales la actividad puede realizarse y Orientación de Interacción de Procesos (PI) es desarrollada desde el punto de vista de las entidades (transacciones) que fluyen en el sistema.

Los ingenieros civiles y constructores utilizan dibujos y gráficas con el objetivo de visualizar los problemas y especificar detalles. Las mallas son una forma gráfica de comunicar conceptos complejos que de otra manera requerirían de una explicación muy larga. (Martínez 1996). Las herramientas basadas en AS hacen uso de mallas que consisten en nodos (actividades y colas) conectados por arcos, que describen el modelo de simulación. La mayoría de herramientas estudiadas a continuación combinan estrategias AS y ES, conocido como las Tres-Fases AS (Tocher and Owen 1960), que distingue entre actividades condicionales y actividades con límites.

General Simulation Program –GSP, (Tocher and Owen 1960), introdujo el concepto de las Tres-Fases AS. El principal objetivo de diseño de GSP fue la eficiencia del tiempo de ejecución (runtime). Como consecuencia un programa escrito en GSP parece sólo cifras, con demasiadas letras como identificadores (Evans 1988). Introdujeron los diagramas de ruedas, como un grupo de cajas unidas por arcos que representa la secuencia de actividades para cada máquina. **BOCUS – Band Of Computer Universal Simulator** (Hill 1971), popularizó el concepto de Diagramas Cíclicos de Actividades (ACD), en los cuales existen círculos (colas) y cajas (actividades) unidos por arcos que significa la secuencia del proceso. Se diferencia con el diagrama de ruedas que las entidades deben alternar entre colas y actividades. **CYCLONE – Cyclic Operation Network** (Halpin & Woodhead 1976), fue especialmente diseñado para simulación de procesos constructivos. Basado puramente en mallas de trabajo por lo cual se considera muy simple. La malla de CYCLONE es una versión desarrollada de ACD, en la cual se modelan actividades condicionales (combi), actividades con límites(normal), colas, contadores y consolidadores. **RESQUE** (Chang 1986) fue diseñado como la continuación de CYCLONE, en donde el modelo no está limitado por la información transmitida por la malla de trabajo. En adición tiene funciones para distinguir y realizar seguimiento de recursos, además de mejorar el control de simulación. **COOPS Sistema de Simulación de Construcción** (Liu 1991), es una extensión del CYCLONE diseñada e implementada usando un lenguaje de programación orientado por objetos. La malla de simulación es una colección de objetos con actividades, colas y arcos dibujados sobre la pantalla. Los recursos específicos son modelados como objetos, posee un calendario para no desarrollar actividades en tiempo de descanso y tiene la capacidad de generar y consolidar recursos en los arcos. **DISCO** (Huang and Halpin 1994), es un pre-proceso y post-proceso a MicroCyclone, como una interfaz gráfica y una animación de simulación respectivamente. **STROBOSCOPE** (Martínez 1996) es un lenguaje de programación que representa los recursos como objetos e identifica propiedades asignables, persistentes y dinámicas; y que puede activa o dinámicamente tomar en consideración el estado del proceso de simulación. Además tiene un diseño abierto para que el modelador escoja cual es la información que entra al modelo (input) y cual es la que desea obtener (output).

La falta de acceso al cuerpo de programación de estas plataformas de simulación como MicroCyclone y de lenguajes de programación para simulación como STROBOSCOPE, ha sido un obstáculo para contar con una herramienta flexible que se adapte a las condiciones de los procesos y de nuestro medio.

MOCSPROC es un MOdelo Computacional de Simulación de PROcesos Constructivos, desarrollado en un lenguaje de programación sencillo y fácil de interpretar como lo es Visual Basic. Además es el primer paso en el desarrollo de una herramienta de simulación de procesos constructivos orientada por eventos y estructurada por módulos, cuyo objetivo es obtener información de la ejecución y comportamiento de procesos constructivos, bajo condiciones y restricciones específicas, y generar reportes sobre estos resultados.

Este artículo presenta a continuación la estructura de simulación de MOCSPROC y explica cada una de sus etapas, para finalmente resaltar las principales conclusiones y recomendaciones que surgieron como consecuencia de la investigación.

ESTRUCTURA DE SIMULACION MOCSPROC

La estructura del modelo computacional está compuesta por 1 formulario y 6 módulos. El formulario, llamado "Simulación", es un objeto que tiene sus propios métodos, propiedades y eventos. El formulario está compuesto por código y por el objeto que se conoce como la interfaz del programa. El código del formulario contiene la secuencia total de simulación que se ejecuta como respuesta al evento de oprimir un control o botón dentro de la interfaz en tiempo de ejecución. En resumen, este formulario es la columna vertebral del modelo computacional porque ejecuta los procedimientos programados en los demás módulos, con un orden lógico y una secuencia definida por el algoritmo de simulación, como respuesta a eventos producidos por el usuario.

Los módulos son simplemente código de programación que contienen declaración de variables, funciones y procedimientos. Estos dos últimos realizan acciones específicas como capturar información del modelo gráfico, desarrollar acciones dentro de la estructura de simulación como revisar condiciones del modelo, programar eventos y permitir el flujo de recursos, modelar duración de actividades, calcular estadísticas y

mostrar resultados.

Por consiguiente los módulos contienen la totalidad de programación del modelo computacional, mientras el formulario utiliza esta programación ordenadamente de acuerdo a los lineamientos de simulación, para obtener el producto final que corresponde al resultado de la simulación y sus estadísticas. La Figura 1 explica gráficamente la estructura de simulación de MOCSPROC.

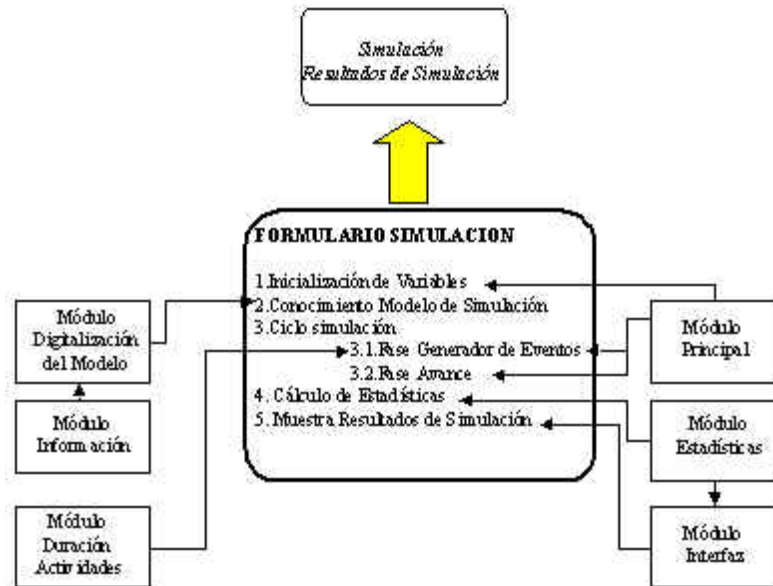


Figura 1. Estructura de Simulación MOCSPROC

INICIALIZACIÓN DE VARIABLES

La primera etapa de la estructura tiene la función de inicializar las variables, contadores y dimensionar las matrices que contienen la información del modelo que se desee modelar y la información de los resultados y cálculo de estadística. Esta etapa se apoya en el Módulo Principal.

CONOCIMIENTO MODELO DE SIMULACION

Esta etapa tiene la función de capturar la información de la red o modelo gráfico de simulación que previamente se ha construido, basado en elementos básicos del sistema de modelación CYCLONE (CYCLic Operations Network), redes que representan ciclos de procesos constructivos (Oglesby 1989). La información de cada uno de los nodos del modelo se copia en las matrices que corresponden a cada tipo de nodo. Entonces existen tantos tipos de matrices como tipos de nodo hay (cola, combi, normal, contador, consolidador). Esta etapa de la estructura se apoya en los módulos Digita Red e Información. El Módulo Digita Red es el único al cual los usuarios tienen acceso, en la Figura 2 se presenta este módulo.

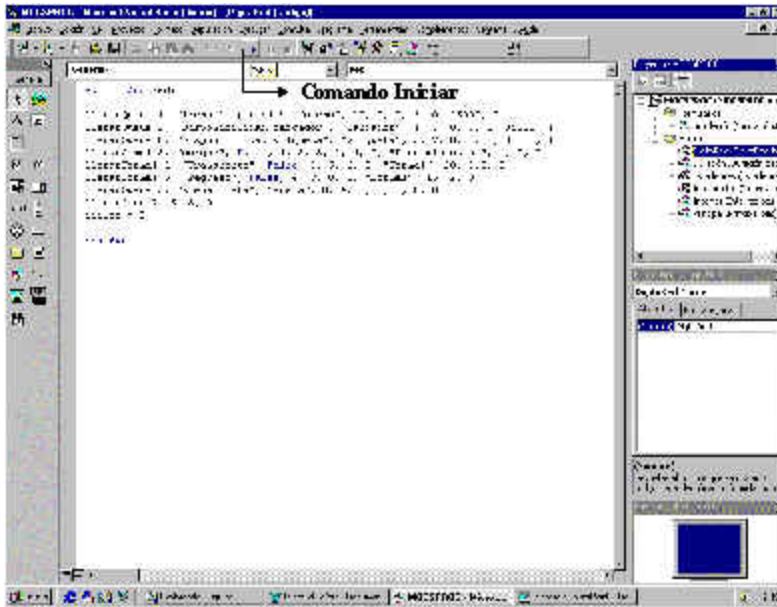


Figura 2. Módulo Digita Red

CICLO DE SIMULACIÓN

La etapa del ciclo de simulación es la más importante dentro de la estructura porque define el proceso básico de simulación MOCSPROC. Está compuesta por la Fase Generadora de Eventos y por la Fase de Avance, las cuales contienen el desarrollo del algoritmo de simulación.

El motor de simulación se basa en el modelo de simulación en red y en las estrategias de simulación: Orientación por Eventos (ES) y Orientación de Seguimiento de Actividades (AS). Su funcionamiento depende de la ocurrencia de eventos discretos que sólo se presentan en tiempos específicos durante el tiempo de simulación, es gobernado por un calendario y ejecución de subrutinas (eventos) y se enfoca en identificar las actividades y las condiciones bajo las cuales las actividades pueden realizarse. No hay distinción entre el flujo de entidades y máquinas; todos son recursos.

El principal concepto que debe tenerse en cuenta para desarrollar un modelo de simulación de procesos, es establecer un método específico que controle eficientemente el tiempo de simulación y determine tiempos exactos para el desarrollo de actividades y eventos que producirán el flujo de recursos.

El concepto de “Reloj de Simulación” hace referencia al control del tiempo de simulación.

En MOCSPROC se emplea un método que se basa en el avance del Reloj de Simulación, de acuerdo con la finalización de las actividades que han sido programadas. Para el diseño de este método, primero se crea una Lista de Eventos que hace las veces de libreta de citas o calendario, en donde se programan la totalidad de actividades que pueden ocurrir y se calcula su tiempo de finalización. Por otra parte se crea la Lista Cronológica, que contiene las actividades que se van ejecutando en total orden. Entonces, cuando los eventos son generados se programan en la Lista de Eventos y éstos eventos son grabados en orden en la Lista Cronológica de acuerdo al momento de finalización.

La Lista de Eventos contiene los eventos que han sido programados, mientras la Lista Cronológica contiene la lista de eventos que se han ejecutado. El último evento de la Lista Cronológica corresponde al evento que se desarrolla actualmente y por consiguiente representa el Tiempo Presente (TNOW).

De esta manera se pueden generar dos fases de simulación, la primera consiste en preguntar dentro del modelo de simulación qué eventos pueden comenzar en el Tiempo Presente (TNOW) y programarlos en la Lista de Eventos; y la segunda consiste en ubicar el evento que se ejecuta primero en la Lista de Eventos, copiarlo en la Lista Cronológica, actualizar el Tiempo Presente (TNOW) y realizar el flujo de

recursos; todo por consecuencia de la ejecución de dicho evento. La transferencia de eventos de la Lista de Eventos a la Lista Cronológica es el mecanismo para el avance del Reloj de Simulación. La última entrada a la Lista Cronológica siempre indica el valor del Tiempo Presente (TNOW) y corresponde siempre al evento más temprano de la Lista de Eventos que no había sido transferido. El algoritmo de simulación se presenta en la Figura 3.

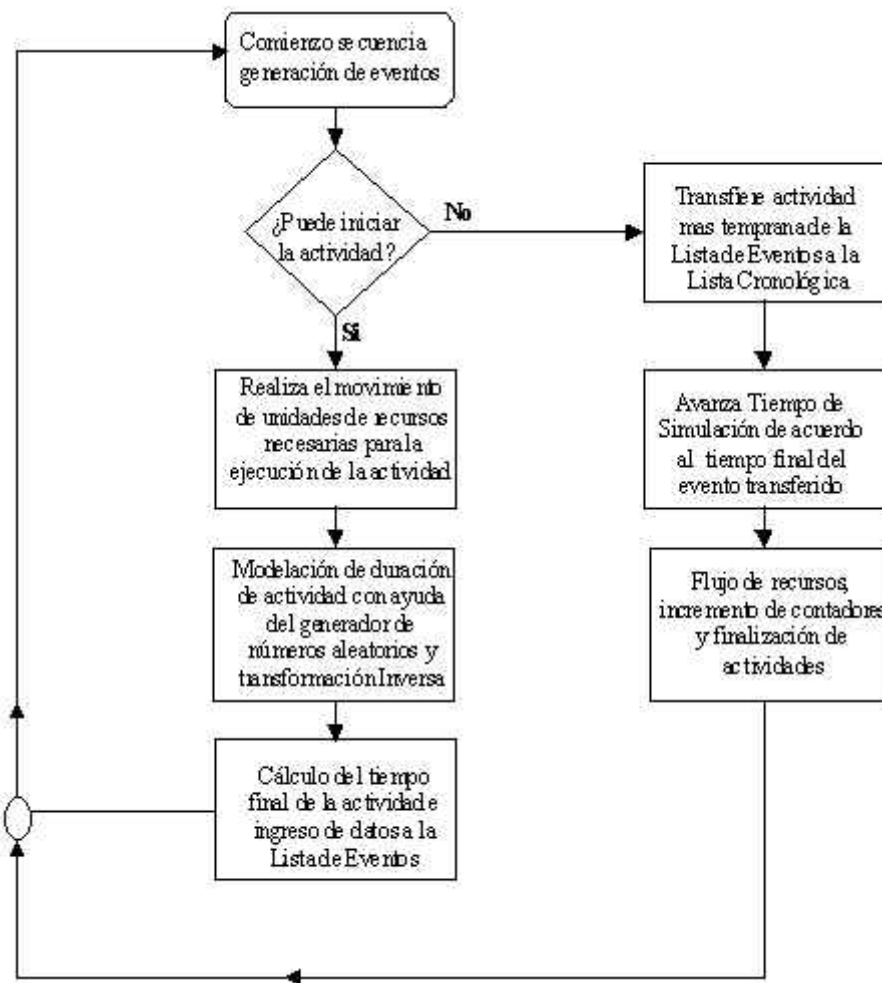


Figura 3. Algoritmo de Simulación

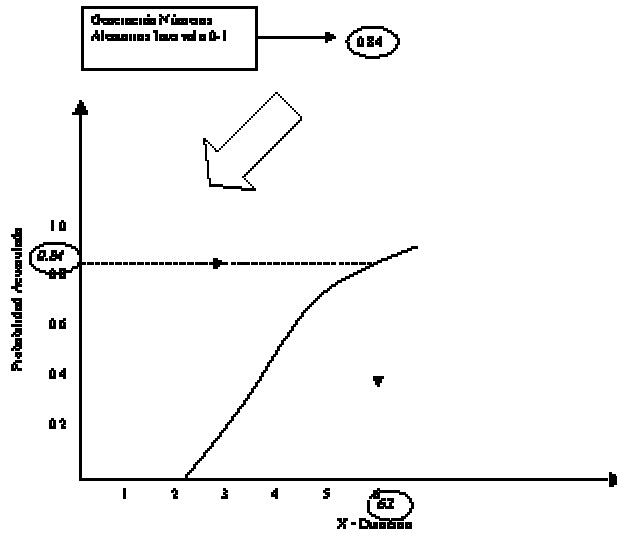
En la primera fase de simulación, Fase Generación de Eventos, se determina si las actividades combi o normal pueden comenzar o no, si es positivo se genera la duración de la actividad correspondiente determinística o probabilística de acuerdo a la descripción del sistema (Simulación Monte Carlo, Módulo Duración), se ocupan los recursos necesarios para la ejecución de la actividad y se ingresa a la Lista de Eventos su código, Tiempo Presente (TNOW), Duración, Tiempo Final y se escribe un "0" que significa que este evento no ha sido transferido a la Lista Cronológica. Por supuesto varias actividades son programadas y se generan varios Tiempos Finales de Eventos. La Figura 4 presenta el proceso de la Simulación Monte Carlo.

Esta fase simplemente programa la iniciación de actividades de acuerdo a las condiciones que presenta la red de modelación en un tiempo determinado.

La Fase de Avance se caracteriza por transferir el evento mas temprano de la Lista de Eventos a la Lista Cronológica. Esto produce que el Tiempo de Simulación avance con el valor del Tiempo Final del Evento transferido, lo que corresponde al Tiempo Presente (TNOW). Entonces, el valor de TNOW siempre corresponde al Tiempo Final del Evento transferido. Cuando el Tiempo de Simulación avanza, todas las actividades que pueden ser finalizadas terminan y se realiza el flujo de recursos respectivo. Después que

termina el flujo de recursos, la Fase Generación de Eventos se inicia de nuevo para iniciar un ciclo hasta cumplir una condición determinada como un Tiempo de Simulación máximo o un número de ciclos máximo.

Se dice que MOCSPROC es una programación discreta por eventos, ya que la generación de eventos depende del flujo de recursos que ocasiona un evento determinado. La etapa de ciclo de simulación se apoya en los módulos Principal y Duración.



Gráfica de la Función de Densidad de Probabilidad Acumulada.

Figura 4. Simulación de Monte Carlo

CÁLCULO DE ESTADÍSTICAS

Esta etapa de la estructura de MOCSPROC está apoyado por el Módulo Estadísticas. Tiene la función de guardar información detallada de las estadísticas de simulación, organizar datos y conformar matrices con el resumen de los resultados más relevantes.

RESULTADOS DE SIMULACIÓN

El Módulo Interfaz apoya esta etapa de la estructura que tiene la función de mostrar los principales resultados de simulación como respuesta a eventos generados por el usuario desde la Interfaz Visual Basic (Figura 5). Estos resultados son presentados en una hoja de Microsoft Excel. Los resultados presentados son:

1. Tiempo total de simulación.
2. Ciclos ejecutados.
3. Lista de Eventos.

4. Lista Cronológica.
5. Tiempo vacante de cada actividad combi.
6. Tiempo vacante de cada actividad normal.
7. Tiempo de permanencia de recursos en las colas.
8. Costo total del proceso.

CONCLUSIONES

Los procesos constructivos son modelados efectivamente utilizando la simulación discreta por eventos, la cual asume que el estado de un sistema cambia instantáneamente en tiempos específicos marcados por eventos. Dentro de las estrategias mas utilizadas para la simulación de procesos constructivos están la Orientación por Eventos (ES) y la Orientación de Seguimiento de Actividades (AS). La estrategia escogida por la herramienta de simulación tiene un impacto alto en la forma que un modelo es presentado en el computador y en cómo el modelador visualiza el mundo real. La mayoría de herramientas computacionales combinan estrategias AS y ES, que utilizan una forma gráfica (mallas) de comunicar conceptos complejos que de otra manera requerirían de una explicación muy larga. El principal concepto que debe tenerse en cuenta para desarrollar un modelo de simulación de procesos, es establecer un método específico que controle eficientemente el tiempo de simulación y determine tiempos exactos para el desarrollo de actividades y eventos que producirán el flujo de recursos.

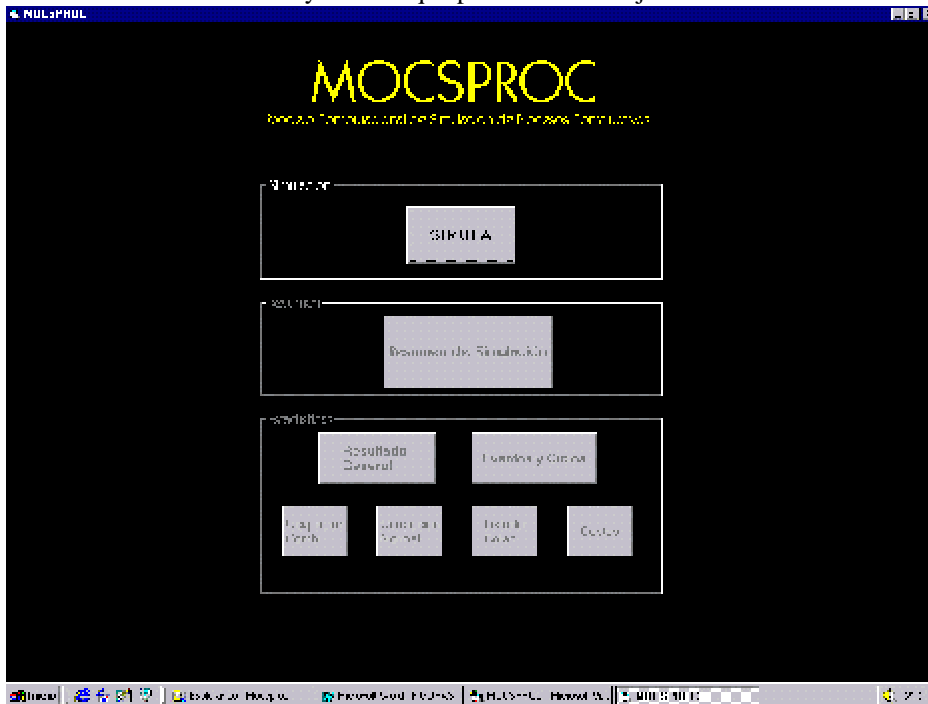


Figura 5. Interfaz MOCSPROC

MOCSPROC (MOdelo Computacional de Simulación de PROCesos Constructivos) basa su comportamiento en la estrategia Tres-Fases AS. Esta estrategia combina la simulación orientada por eventos y el seguimiento de actividades, verifica condiciones bajo las cuales pueden desarrollarse actividades específicas y distingue entre actividades condicionales (Combi) y actividades con límites (Normal). Su funcionamiento depende de la ocurrencia de eventos discretos que sólo se presentan en tiempos específicos durante el tiempo de simulación, es gobernado por un calendario y ejecución de subrutinas (eventos) y se enfoca en identificar las actividades y las condiciones bajo las cuales las actividades pueden realizarse. No hay distinción entre el flujo de entidades y máquinas; todos son recursos. De esta manera se pueden generar dos fases de simulación, la primera consiste en preguntar dentro del modelo de simulación qué eventos pueden comenzar en el Tiempo Presente (TNOW) y la segunda consiste en ubicar el evento que se ejecuta primero, actualizar el Tiempo Presente (TNOW) y realizar el flujo de recursos.

MOCSPROC se consolida como un modelo computacional estructurado por módulos, programado en un lenguaje sencillo y flexible, capaz de simular procesos constructivos complejos y abierto a cualquier tipo de cambios que nuevos programadores deseen realizar para optimizar su proceso de simulación.

Es importante que para próximos trabajos de investigación de simulación de procesos constructivos se desarrolle más profundamente la interfaz de MOCSPROC, con el propósito de arreglar su fracturación actual y ofrecer al usuario opciones más didácticas. Además se recomienda optimizar la información solicitada al usuario, ya que existen datos que en realidad no se utilizan en el proceso de simulación. También es conveniente retomar el proceso de validación de este modelo computacional para ubicar puntos críticos que permitan desarrollar cada vez más, una herramienta que se adapte a condiciones generales del medio.

Finalmente, es importante crear un ambiente propicio para que MOCSPROC se convierta en el punto de partida de futuras investigaciones avanzadas en temas de simulación de procesos constructivos.

REFERENCIAS BIBLIOGRÁFICAS

American Society of Civil Engineers, *Journal of Computing in Civil Engineering / Application Framework for*

Development of Simulation Tools, Ed. William J. Rasdorf, Vol.14 N.3, (July 2000).

Echeverry, Diego, *Notas de Clase Herramientas Computacionales*, Universidad de los Andes, 2000.

Gallardo Eraso, Rafael Hernando, *Modelación y Simulación de Procesos Constructivos para el Mejoramiento*

de la Productividad, Tesis Universidad de los Andes, 1999.

Halpin, Daniel W. & Leland S Riggs, *Planning and Analysis of Construction Operations*. John Wiley & Sons,

Inc., New York, (1992)

Hodge, Charles S., *Computing in Civil Engineering*, American Society of Civil Engineers, New York (1984)

Kalos, Malvin H., Whitlock, Paula A, *Monte Carlo Methods*, John Wiley & Sons, Inc., New York, (1986)

Martínez, Julio, *STROBOSCOPE State and Resource Based Simulation of Construction Processes*, University

of Michigan, 1996

Microsoft Corporation, *Visual Basic 6.0 Manual del Programador*, McGraw Hill, Madrid, 1998.

Oglesby, C.H., H.W. Parker, and G. A. Howell, *Productivity Improvement in Construction*. McGraw-Hill,

Inc., New York, N.Y. (1989).

Pritsker, A Alan B & Pegden, Claude Dennis, *Introduction Simulation and Slam*. John Wiley & Sons, New

York, 1979.