8

Una herramienta computacional gráfica para el modelado y la simulación dinámica de sistemas informáticos

Jorge Rubén Vega

Se presenta una herramienta computacional de carácter gráfico que permite modelar un sistema informático (SI) y simular su comportamiento dinámico. La herramienta está compuesta por tres clases de módulos: de generación, de atención, o de encaminamiento de trabajos. Los diferentes módulos intercambian flujos de trabajo, y se comportan como centros dinámicos o estáticos. Los centros dinámicos se modelan matemáticamente por medio de ecuaciones diferenciales ordinarias; y los centros estáticos a través de ecuaciones algebraicas. La estructura gráfico/modular con que se dota a cada centro permite su interconexión para construir módulos de mayor jerarquía, destinados a modelar componentes o dispositivos de un SI más complejo. La metodología de modelado y simulación propuesta permite predecir la evolución temporal de las principales variables asociadas con cada dispositivo del SI, y provee resultados de estado estacionario coincidentes con los obtenidos a través de algoritmos clásicos. La herramienta es útil para evaluar la «performance» de un SI, y es fácil de utilizar en un curso universitario de grado.

Jorge Rubén Vega
Cargo/Institución:Prof. Adjunto Ordinario (Univ. Tecnológica Nacional - Fac. Regional Santa Fe)
Investigador Asistente (CONICET - INTEC)
Dirección Postal: Lavaise 610 (3000) Santa Fe - Argentina
jvega@arcride.edu.ar

Proyecto Marco: «Análisis y Evaluación del Comportamiento de una Red Informática». Proyecto 25/O046. (Grupo de Investigación en Redes - GIRED). Fac. Regional Santa Fe (UTN). Directora: Dra. A.R. Tymoschuk.

GIRED - Facultad Regional Santa Fe - Universidad Tecnológica Nacional. Lavaise 610 (3000) Santa Fe - Argentina
E-mail: jvega@arcride.edu.ar

Introducción

Debido al gran avance de los sistemas de comunicaciones y a su uso masivo e intensivo, la complejidad de los sistemas informáticos (SI) ha crecido notoriamente a lo largo del tiempo, tanto desde el punto de vista del hardware como del software utilizado. Actualmente, en lapsos de tiempo relativamente cortos se requiere actualizar y modificar estos sistemas, en busca de satisfacer la demanda pretenciosa de los usuarios: lograr tiempos de respuesta cada vez más pequeños para manipular transacciones con cargas de trabajo cada vez más elevadas. Indudablemente, para conseguir estas mejoras no basta sólo con disponer de un SI moderno, con componentes cada vez más veloces; sino que también es importante administrar adecuadamente los recursos disponibles, evaluar su prestación, sus debilidades o cuellos de botella, sus puntos de posible mejora, etc. En tal sentido, la evaluación de medidas de «performance» de un SI es una disciplina de gran importancia; y la simulación numérica de un modelo puede prestar una gran ayuda para analizar potenciales modificaciones o rediseños del SI existente.

Desafortunadamente, el desarrollo de un modelo representativo de un SI suele ser una tarea compleja, dado que las variables involucradas

(continuas o discretas) evolucionan en el tiempo y tienen normalmente atributos aleatorios. Estrictamente, deberían utilizarse modelos derivados de la teoría general de los procesos estocásticos; pero la complejidad de este tipo de modelos obliga muchas veces a recurrir a importantes simplificaciones. Los procesos de Markov, la teoría de colas, y especialmente los modelos de redes de cola, son posiblemente las herramientas teóricas más utilizadas para el modelado matemático de SI (Lazowska y col., 1984; Conway y Georganas, 1989; Puigjaner y col., 1992). Aplicaciones de esas herramientas destinadas a la evaluación de la «performance» de redes de computadoras pueden consultarse en Kleinrock (1993), y Menascé y Almeida (1998). Más recientemente, aplicaciones en el ámbito de los negocios electrónicos fueron publicadas por Menascé y Almeida (2000). En las últimas décadas han surgido otros enfoques alternativos, líneas de investigación y herramientas para el modelado de SI, como por ejemplo las redes de Petri (Murata, 1989; Reutenauer, 1990), y la teoría de los sistemas a eventos discretos (Cohen y col., 1989; Banks y col., 1996), pero no serán aquí tratados.

Las características dinámicas de los SI son innegables; y estrictamente, un modelo riguroso debería permitir la predicción de la evolución temporal de las variables involucradas. La dinámica de los sistemas a eventos discretos ha sido discutida en la literatura, y comparada con la dinámica de los sistemas continuos (Ho, 1989; Cao, 1989), aunque la mayoría de los modelos matemáticos desarrollados para SI normalmente no la contemplan. Por el contrario, el análisis de la «performance» de un SI suele efectuarse en base a un modelo operacional, matemáticamente representado por expresiones algebraicas recursivas derivadas de la teoría de colas. La resolución de dichas expresiones se efectúa por medio de algoritmos algebraicos que resuelven el problema estacionario. Entre los algoritmos típicamente utilizados, podemos mencionar los denominados MVA («mean value analysis») exactos y aproximados (Lazowska y col., 1984). Estos algoritmos son relativamente sencillos de implementar (basta una simple planilla de cálculo), pero adolecen de al menos dos limitaciones importantes: a) su complejidad conceptual para interpretar el significado físico de cada uno de sus pasos; y b) sólo proveen resultados de estado estacionario.

Un conocimiento exhaustivo sobre modelado matemático de SI escapa usualmente a la formación académica recibida por un estudiante de grado; y también -posiblemente- a la preparación media de un profesional de la ingeniería informática. Sería de interés entonces disponer de alguna herramienta que permita con cierta facilidad modelar un SI y evaluar su «performance», logrando predicciones razonablemente aproximadas, aun a costa de sacrificar el rigor científico que proveerían los modelos más precisos. En este trabajo, se propone una metodología para modelar y simular el comportamiento dinámico de un SI. La idea general perseguida es desarrollar una herramienta computacional que permita en forma sencilla modelar un SI, analizar su funcionamiento, practicar modificaciones en su configuración, determinar medidas de «performance», identificar dispositivos conflictivos, etc. Además, resultará de interés proveer a esta herramienta de: a) un carácter gráfico/modular; y b) una capacidad de modelado jerárquico.

Consideraciones preliminares

En general, cualquier SI puede representarse mediante la interconexión de un conjunto de dispositivos -a los que genéricamente denominaremos módulos-, capacitados para recibir, generar (o consumir), atender, encaminar, y/o transferir trabajos de cómputo. En particular, denominaremos centros a los módulos más sim-

ples (o de menor jerarquía) que componen el SI; y se caracterizarán por manipular una única clase o tipo de trabajos. La combinación de dos o más centros dará lugar a la formación de un módulo de mayor jerarquía.

Según su función en el SI, los centros pueden comportarse en forma dinámica o estática. Un centro dinámico (CD) es un conjunto de dispositivos con capacidad para almacenar trabajos, a los que procesa y/o demora. En el caso más general, un CD puede además generar, consumir, recibir y/o entregar un flujo de trabajos. En la Figura 1, se representa en forma esquemática un CD 'k'. Nótese que las variables representativas del «flujo de información» entre los centros (uk y xk), son indicadas en líneas más gruesas; mientras que Qk es sólo una «salida» del bloque disponible para capturar su valor, sin que haya por ella ningún flujo de información (se la representa con una línea más fina).

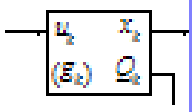
Admitiremos que todas las variables de un CD evolucionan en forma continua en el tiempo (t). En un intervalo de tiempo suficientemente pequeño, un balance global de los trabajos en el CD puede expresarse cualitativamente de la siguiente manera:

$$\begin{bmatrix}\text{Variación temporal}\\ \text{de trabajos en el CD}\end{bmatrix} = \begin{bmatrix}\text{Flujo de ingreso}\\ \text{de trabajos}\end{bmatrix} - \begin{bmatrix}\text{Flujo de egreso}\\ \text{de trabajos}\end{bmatrix} + \begin{bmatrix}\text{Flujo de generación}\\ \text{(consumo) de trab.}\end{bmatrix}$$

Este balance se representa a través de la siguiente ecuación diferencial ordinaria:

$$\frac{dQ_k(t)}{dt} = u_k(t) - x_k(t) + g_k(t) \;\; ; \quad Q_k(t_0) = Q_{k,0} \qquad (1)$$

donde Qk,0 representa al número de trabajos en el CD, en un dado tiempo t0. (Nótese que gk será positivo si se generan trabajos, y negativo si se consumen o pierden.)

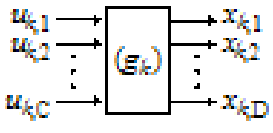


Variables del CD 'k'

Nombre	Descripción
u_k [trab./s]	Flujo de ingreso de trabajos
g_k [trab./s]	Velocidad de generación (o consumo) de trabajos
Q_k [trab.]	Número de trabajos en el CD
x_k [trab./s]	Flujo de egreso de trabajos

Figura 1: Esquema modular de un CD, y descripción de sus variables asociadas.

Un centro estático (CE) es un conjunto de dispositivos capacitado para recibir y/o entregar información en un tiempo infinitamente pequeño (en la práctica, mucho menor que la menor constante de tiempo del SI). Desde este punto de vista, un CE puede ser considerado como un CD ficticio, con una dinámica infinitamente rápida. En el caso más general, un CE puede recibir C flujos de entrada, entregar D flujos de salida, y generar (o consumir) trabajos. En la Figura 2, se representa en forma esquemática un CE 'k'. Admitiremos que todas las variables asociadas con un CE evolucionan en forma continua en el tiempo.



Variables del CE 'k'

Nombre	Descripción
$u_{k,i}$ [trab./s]	Flujos de ingreso de trabajos (i=1,...,C)
g_k [trab./s]	Flujo de generación (o consumo) de trabajos
$x_{k,i}$ [trab./s]	Flujos de egreso de trabajos (i=1,...,D)

Figura 2: Esquema modular de un CE, y descripción de sus variables asociadas.

Los CE se encuentran permanentemente en estado estacionario, y entonces el balance de la ec (1) se reduce a:

$$0 = \sum_{i=1}^{c} u_{ej}(t) - \sum_{i=1}^{b} x_{ej}(t) + g_e(t) \qquad (2)$$

En las dos secciones siguientes, se presentan las características operativas y los modelos matemáticos propuestos para diferentes tipos de CD y de CE.

Modelado matemático de los centros dinámicos

El funcionamiento de cada CD se describirá mediante un modelo matemático determinístico de tiempo continuo, basado en la ec. (1). Para su resolución, es necesario proponer una relación que permita calcular la velocidad de procesamiento xk, la que dependerá del tipo de CD en cuestión. En este trabajo, consideraremos dos clases de CD: los centros de cola, y los centros de generación por terminales.

Centro de Cola

Un centro de cola (CC) es un CD en el que mientras algunos trabajos están siendo atendidos o procesados, los restantes esperan en cola su turno. Ejemplos de CC son: una CPU, un sistema de discos rígidos, un canal de transmisión de datos, etc. En una primera instancia, se admitirá que en un CC todos los trabajos son similares, y requerirán entonces el mismo tiempo medio de servicio o atención (Sk = cte.). En la Figura 3.a), se representa un esquema operacional de un CC, típicamente utilizado para diferenciar los trabajos en servicio de los ubicados en cola de espera. Sin embargo, debe notarse que con relación al modelo matemático sólo interesará computar el número total de trabajos en el CC (Qk).

Veamos ahora cómo evaluar la velocidad de procesamiento de trabajos, xk, en un CC. Intuitivamente, xk depende al menos de: i) el tiempo medio de servicio, Sk; ii) la cantidad de trabajos en el CC, Qk; y iii) la cantidad total de trabajos residentes en el SI, N. Por un lado, es bastante evidente que xk será inversamente proporcional a Sk. Por otra parte, es de esperar que xk sea proporcional a Qk, dado que al aumentar Qk mayor utilización tendrá el CC (hay más disponibilidad de trabajos para ser atendidos y resueltos); pero los incrementos marginales de xk con Qk serán cada vez menores, porque el CC tenderá gradualmente a saturarse. Por último, la dependencia de xk con N es más indirecta, porque no se puede considerar al CC en forma aislada, dado que los N trabajos se reparten entre todos los recursos del SI. En general, al aumentar N, se espera una mayor utilización de cada uno de los recursos, y un incremento de xk como consecuencia de un aumento en Qk.

No existe una expresión analítica única que permita calcular fielmente la velocidad de procesamiento xk en un CC. En este trabajo se propone que la expresión a utilizar satisfaga los requerimientos comentados en el párrafo anterior, y además que permita reproducir los valores estacionarios provistos por la técnica MVA aproximada. Concretamente, se propone calcular xk (t) a través de la siguiente expresión:

$$x_k(t) = \frac{Q_k(t)}{R_k(S_k, Q_k, N)} = \frac{Q_k(t)}{S_k\left[1 + Q_k(t) - \frac{Q_k(t)}{N}\right]} \qquad (3)$$

donde Rk es un tiempo de residencia característico por cada visita de un trabajo al CC. Nótese que si N = 1, entonces Rk = Sk; y si N ⊗ (, resulta Rk = Sk + Sk Qk. La ec. (3) es exacta para N = 1, y asintóticamente correcta para valores elevados de

N (Lazowska y col., 1984). Reemplazando la ecn. (3) en la (1), resulta:

$$\frac{dQ_k}{dt} = u_k(t) - \frac{Q_k(t)}{S_k\left[1 + Q_k(t) - \frac{Q_k(t)}{N}\right]} + g_k(t) \; ; \quad Q_k(t_0) = Q_{k,0} \qquad (4)$$

Esta expresión admite la representación modular de la Fig. 3.b).

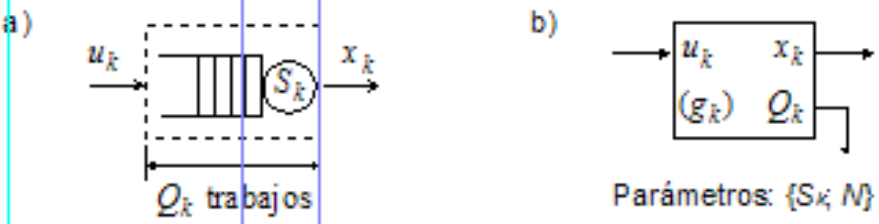


Figura 3: Representación de un CC: a) esquema operacional; b) estructura modular.

La ecn. (4) puede extenderse al caso de un CC que atienda CL clases de trabajos de distinto requerimiento medio (clases múltiples). El tiempo característico para la clase 'c', se calculará como:

$$R_{k,c}(t) = S_{k,c}\left[1 + \sum_{j=1}^{CL} Q_{k,j}(t) - \frac{Q_{k,c}(t)}{N_c}\right] \; ; \quad (c = 1, \ldots, CL) \qquad (5)$$

donde Sk,c y Qk,c son el tiempo de servicio y el número de trabajos de la clase c en el CC 'k', respectivamente. Entonces, para el CC 'k' deberá plantearse una EDO por cada clase:

$$\frac{dQ_{k,c}}{dt} = u_{k,c}(t) - \frac{Q_{k,c}(t)}{S_{k,c}\left[1 + \sum_{j=1}^{CL} Q_{k,j}(t) - \frac{Q_{k,c}(t)}{N_c}\right]} + g_{k,c}(t) \; ; \quad \begin{matrix} Q_{k,c}(t_0) = Q_{k,c,0} \\ (c = 1, \ldots, CL) \end{matrix} \qquad (6)$$

donde uk,c y gk,c son los flujos de trabajo de clase c; y Nc es el número de trabajos de clase c. Nótese que para una única clase de trabajos (CL=1), la ec. (6) se reduce a la ec. (4).

Centro de Generación por Terminales

En un centro de generación por terminales (CGT), cada usuario (ubicado en una terminal) origina un trabajo nuevo para el SI, cuando recibe la notificación de que ha concluido el procesamiento de su último trabajo enviado. Entonces, la velocidad de generación de nuevos trabajos (gT) es igual a la velocidad de notificación de trabajos concluidos (μT), mientras que la velocidad de arribo de trabajos es nula. Supongamos que hay NT terminales activas, y que Z es el tiempo medio de pensado o reflexión utilizado por cada usuario para decidir la generación de un nuevo trabajo. Entonces, el CGT puede ser considerado como un CC con Qk = QT, y Rk = Z. En tal caso, la velocidad de «procesamiento» en las terminales se modela como xT = QT / Z; y a partir de la ecn. (1), resulta:

$$\frac{dQ_T}{dt} = u_T(t) - \frac{Q_T(t)}{Z} + \delta_i N_{T,i} \; ; \quad Q_T(t_i) = N_{T,0} \qquad (7)$$

donde QT(t0)=NT,0 indica que, a t=t0, todos los trabajos residen en las terminales antes de ser remitidos para su procesamiento. El término (i NT,i contabiliza el ingreso de NT,i nuevas terminales en el tiempo t=ti, y (i es una «delta de Dirac» en t=ti (la baja de un usuario se representa por un NT,i<0). Debe notarse que en un

CGT QT(t)(NT(t). En la Fig. 4, se presentan el modelo operacional y el esquema modular de un CGT.

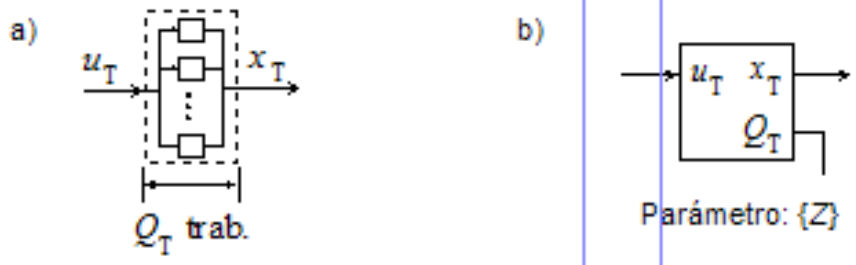


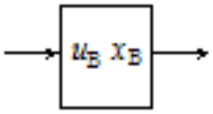
Figura 4: Representación de un CGT: a) esquema operacional; b) estructura modular

Modelado matemático de los centros estáticos

El funcionamiento de un CE se describirá mediante una expresión algebraica basada en la ec. (2). En este trabajo, consideraremos tres clases de CE: los centros de generación «batch», los nodos de confluencia y los nodos de derivación.

Centro de Generación «batch»

Un centro de generación «batch» (CGB) puede interpretarse como una cola de trabajos que están permanentemente disponibles por el SI. Dicha cola provee en forma instantánea: a) un trabajo nuevo por cada trabajo resuelto en el SI; y b) NB nuevos trabajos, al aumentarse en NB el grado de multiprogramación del SI. En la Fig. 5, se presenta el esquema modular de un CGB. El modelo matemático de la ec. (8) se obtuvo a partir de la ec. (2), con C=D=1, y gB(t)=(i NB,i . En esta expresión NB,i es el cambio del grado de multiprogramación en el tiempo ti; uB(t) es la velocidad de procesamiento de trabajos «batch» en el SI; y xB(t) es la velocidad de aporte de nuevos trabajos al SI. Nótese que, xB(t) = uB(t), (t (ti.



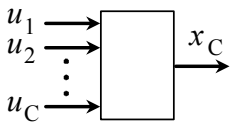
$$x_B(t) = u_B(t) + \delta N_{B,i} \qquad (8)$$

Figura 5: Estructura modular y modelo matemático de un CGB.

Nodos de Confluencia y de Derivación de Trabajos

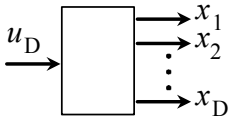
Un nodo de confluencia de trabajos (NCT) es un punto de un SI al que concurren simultáneamente C (>2) flujos de trabajos, ui(t), i=1,..., C; y del que parte un único flujo, xC(t). Un nodo de derivación de trabajos (NDT) es un punto del SI al que arriba un único flujo de trabajos, uD(t), y del que parten simultáneamente D (>2) flujos de trabajos, xi(t), i=1,..., D. En cualquiera de estos nodos admitiremos que: 1) no se generan ni se destruyen o pierden trabajos (g=0); y 2) el encaminamiento de esa información se efectúa en forma instantánea, sin acumularse trabajos (CE). En un NDT se requiere conocer en que proporción se encamina el flujo de trabajos hacia cada una de las D salidas. Llamaremos pi(t), i=1,...D, a la probabilidad instantánea de que un trabajo que arriba al NDT se encamine hacia la salida 'i'. En la Fig. 6, se muestran los esquemas modulares de ambos nodos, y sus correspondientes modelos matemáticos derivados a partir de la ec. (2). Nótese que: 1) D=1, en un NCT; 2) C=1, en un NDT; y 3) el módulo de un NCT no posee parámetros internos.

NCT:



$$x_C(t) = \sum_{i=1}^{C} u_i(t) \qquad (9)$$

NDT:



$$x_i(t) = p_i(t)\, u_D(t) \;; \quad (i = 1, ..., D) \qquad (10.a)$$

$$\sum_{i=1}^{D} p_i(t) = 1 \qquad (10.b)$$

Parámetros: $\{p_1,..., p_D\}$

Figura 6: Estructura modular y modelo matemático de los NCT y NDT.

Ejemplos de aplicación

Se construyó una «biblioteca» de centros básicos para trabajos de clase simple, con los siguientes componentes: 1) un CC [ec. (4)]; 2) un CGT [ec. (7)]; 3) un CGB [ec. (8)]; 4) un NCT [ec. (9)]; y 5) un NDT [ec. (10)]. Estos centros permiten construir módulos más complejos y SI. A efectos de mostrar la metodología propuesta, se presentan a continuación algunos ejemplos tomados de la literatura (Lazowska y col., 1984). Los modelos matemáticos propuestos se implementaron en la interface gráfica Simulink, de MatlabTM.

Ejemplo 1. Un servidor central (1 CPU / 2 discos) es accedido desde 3 terminales, con trabajos de requerimientos similares (clase simple). En la Fig. 7.a) se muestra la configuración detallada del SI. En el módulo jerárquico desarrollado para el servidor central (Fig. 7.b), se seleccionaron arbitrariamente algunas variables internas para disponerlas como variables de salida; pero sólo la velocidad de procesamiento global (X) podrá utilizarse para la interconexión con otros módulos (Fig. 7.c). A partir de las salidas del modelo, pueden calcularse otras medidas de «performance» (por ejemplo, las velocidades de procesamiento en la CPU y en los discos, sus tiempos de respuesta, etc.).

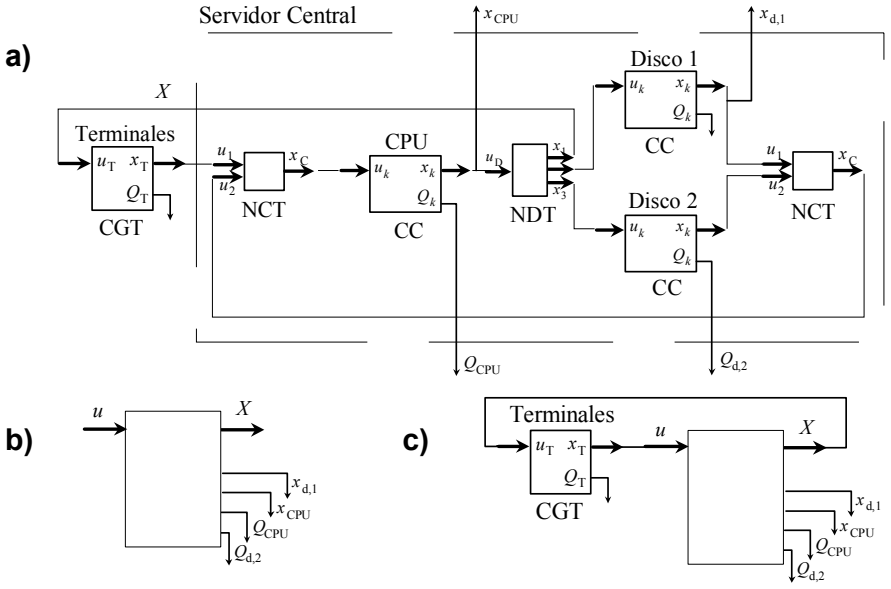


Figura 7. Ejemplo 1: a) esquema modular detallado; b) módulo jerárquico del servidor central (una clase de trabajos); y c) esquema jerárquico equivalente.

En la Tabla 1, se presentan los datos del ejemplo, y se comparan los resultados estacionarios de algunas variables obtenidos aplicando la metodología propuesta con los resultados provistos por los algoritmos MVA exacto y aproximado. Las probabilidades del NDT se calcularon en base a las visitas de los trabajos a los distintos recursos, según: p1 = 1/VCPU = 1/121; p2 = Vd,1/VCPU = 70/121; p3 = Vd,2/VCPU = 50/121. El tiempo total de respuesta R se calculó por aplicación directa de la ley de Little (R=NT/X-Z). Nótese que los estados estacionarios provistos por el modelo coinciden con las predicciones del algoritmo MVA aproximado. En la Fig. 8, se muestran las evoluciones temporales de las colas de trabajo y de las velocidades de procesamiento en cada CC y en el CGT, correspondientes al proceso de arranque del sistema. Al inicio del proceso (t»0), todos los trabajos están en las terminales (QT»3); y luego se va cargando cada dispositivo hasta alcanzar el estado estacionario, después de unos 20 s.

Tabla 1: Datos y resultados de estado estacionario del Ejemplo 1

Datos

S_{CPU}=5 ms	$S_{d,1}$=30 ms	$S_{d,2}$=27 ms	Z=15 s
V_{CPU}=121	$V_{d,1}$=70	$V_{d,2}$=50	N_T=3

Resultados de Estado Estacionario

	Q_T [tr]	Q_{CPU}[tr]	$Q_{d,1}$[tr]	$Q_{d,2}$[tr]	x_{CPU} [tr/s]	X[tr/s]	R[s]
Método Propuesto	2.265	0.0972	0.4021	0.2359	18.26	0.1509	4.881
MVA Exacto [*]	2.273	0.0976	0.3947	0.2350	18.39	0.1520	4.737
MVA Aprox. [*]	2.265	0.0972	0.4021	0.2359	18.26	0.1509	4.881

[*] Resultados extraídos de Lazowska y col. (1984).

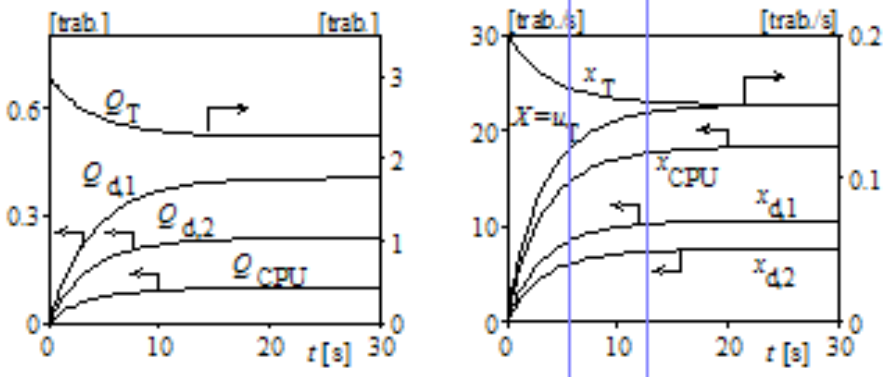


Figura 8. Evoluciones temporales de algunas variables del SI del Ejemplo 1.

Ejemplo 2. Un servidor (1 CPU / 1 disco) atiende dos clases de trabajos (A y B), que arriban al SI como carga transaccional con uA=3/19 trab./s, y uB=2/19 trab/s. Para cada clase de trabajo, los tiempos de servicio y las visitas a cada recurso son: SA,CPU=1/10 s; SA,d=1/3 s; SB,CPU=2/5 s; SB,d=1 s; VA,CPU=10; VA,d=9; VB,CPU=5; VB,d=4. Las probabilidades de los NDT se calcularon como: p1,A=1/10; p2,A=9/10; p1,B=1/5; y p2,B=4/5. Utilizando la ec. (6) con CL=2 (c=A, B), se construyó un módulo de CC para dos clases de trabajo, y se lo utilizó para representar tanto a la CPU como al disco. Se adoptaron como variables de salida las colas de trabajos en cada recurso y la velocidad de procesamiento, para cada clase. En la Fig. 9, se muestra el esquema modular utilizado, y el módulo jerárquico posteriormente construido para un servidor central con una

CPU y un disco, y para dos clases de trabajos.

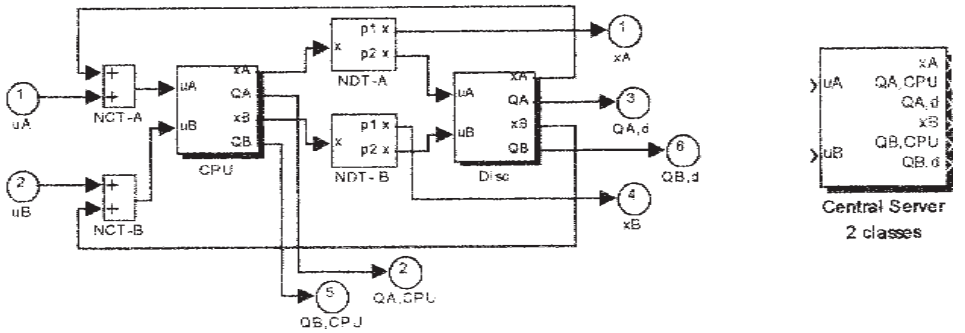


Figura 9. Esquema modular para la simulación numérica del Ejemplo 2, y módulo jerárquico equivalente del servidor central con dos clases de trabajos.

Los resultados dinámicos se presentan en la Fig. 10. Puede observarse los elevados tiempos requeridos para la estabilización de las colas de trabajo en los discos (> 500 s), con respecto a las otras variables del SI. Los valores estacionarios coinciden con los resultados obtenidos por el algoritmo MVA para sistemas abiertos con clases múltiples. Como era de esperar, al alcanzarse el estado estacionario se verifica: xA ® uA; y xB ® uB.

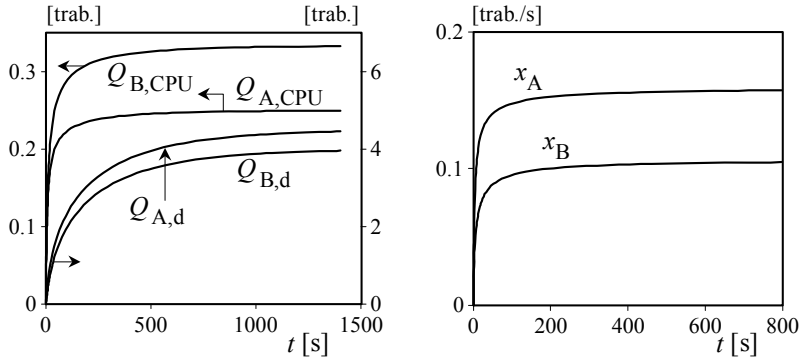


Figura 10. Evoluciones dinámicas de algunas variables del SI del Ejemplo 2.

Ejemplo 3. Se utiliza el servidor del Ejemplo 2 para atender dos colas de trabajos «batch» (clases A y B), con NA=NB=1. Después de los primeros 4 s, se incrementa en uno el grado de multiprogramación de la clase A (NA=2). En la Fig 11, se muestra el esquema modular utilizado para la simulación, y los resultados dinámicos correspondientes. La discontinuidad observada a t=4 s en QA,CPU corresponde al ingreso del nuevo trabajo de la clase A directamente a la cola de la CPU. La variable de estado estacionario más afectada después del ingreso del nuevo trabajo es el número de trabajos en cola en el disco, correspondiente a la clase A (QA,d). Por otra parte, puede observarse que los tiempos de estabilización de todas las variables en el caso «batch», resultaron notoriamente inferiores a los observados en la operación transaccional (Ejemplo 2). En efecto, dicha operación resultó más exigente como consecuencia del mayor número de trabajos («9) residentes en el SI.

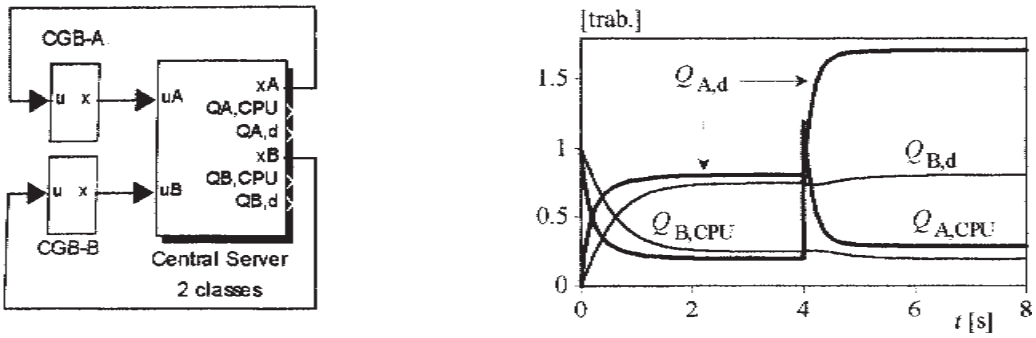


Figura 11. Esquema modular y simulación numérica dinámica del Ejemplo 3.

Discusión final

Se desarrollaron modelos matemáticos (continuos y determinísticos) para describir el funcionamiento de algunos dispositivos básicos, comúnmente utilizados en la descripción de un SI. La validez de los modelos parece más justificada en un SI con un nivel de carga elevado; porque en tal caso, el funcionamiento de cada dispositivo puede aproximarse mejor a través de variables continuas en el tiempo. Pero, independientemente del nivel de carga, la tendencia asintótica de las ecuaciones provee resultados de estado estacionario coincidentes con los obtenidos a partir de los algoritmos clásicos.

El carácter gráfico/modular de la herramienta propuesta facilita la construcción del modelo de simulación de un SI. La variable de interconexión de los módulos es el flujo de trabajos. Sin embargo, al desarrollar diferentes módulos con el objeto de confeccionar una «biblioteca» de dispositivos, es conveniente dejar disponibles otras variables de salida (típicamente, las colas de trabajo), para posibilitar la posterior evaluación de las medidas de «performance».

La capacidad jerárquica de la herramienta posibilita evaluar un SI de mayor complejidad, como puede ser una red informática. A tales efectos, se requerirá desarrollar nuevos módulos para dispositivos típicos de red (por ej., un «router»); módulos que tengan en cuenta efectos no contemplados (por ej., manipulación de trabajos en cola con prioridad); etc.

Las características dinámicas de la herramienta permiten detectar comportamientos críticos del SI durante un transitorio de magnitud importante (como podría ser el provocada por la salida de servicio de un dispositivo en una red), que quizás no sería detectado en un análisis estacionario (por ej., una oscilación o un sobrevalor de una variable que produzca la saturación temporaria de un dispositivo). También, podrían evaluarse políticas de control y optimización dinámica, manipulando tráfico y/o encaminamiento de cargas de trabajos, con el objetivo de mejorar la «performance» global del SI. En la tal caso, se requerirá probablemente definir nuevas métricas de «performance» dinámica.

Por último, la sencillez de la herramienta posibilita su utilización en cursos universitarios de grado, dedicados al estudio de SI y a la determinación de sus medidas de «performance».

NOTA: tanto la biblioteca de módulos básicos como los ejemplos presentados en este trabajo pueden solicitarse al autor por correo electrónico.

Referencias bibliográficas

BANKS, Jerry; CARSON, John; NELSON, Barry (1996). Discrete-Event System Simulation (Second Edition). Prentice Hall. New Jersey.

CAO, Xi (1989). A Comparison of the Dynamics of Continuous and Discrete Event Systems. Proc. IEEE 77(1): 7-13.

COHEN, Guy; MOLLER, Pierre; QUADRAT, Jean; VIOT, Michel (1989). Algebraic Tools for the Performance Evaluation of Discrete Event Systems. Proc. IEEE 77(1): 39-58.

CONWAY, Adrian; GEORGANAS, Nicolas (1989). Queueing Networks. Exact Computational Algorithms. The MIT Press. Massachusetts.

HO, Yu (1989). Dynamics of Discrete Event Systems. Proc. IEEE 77(1): 3-6.

KLEINROCK, Leonard (1993). On the Modeling and Analysis of Computer Networks. Proc. IEEE 81(8): 1179-1191.

LAZOWSKA, Edward; ZAHORJAN, John; GRAHAM, Scott; SEVCIK, Kenneth (1984). Quantitative System Performance. Computer System Analysis Using Queueing Network Models. Prentice Hall, Inc. Englewood Cliffs, New Jersey.

MENASCÉ, Daniel; ALMEIDA, Virgilio (1998). Capacity Planning for Web Performance. Metrics, Models, and Methods. Prentice Hall, Inc. Englewood Cliffs, New Jersey.

MENASCÉ, Daniel; ALMEIDA, Virgilio (2000). Scaling for E-Business. Technologies, Models, Performance, and Capacity Planning. Prentice Hall, Inc. Upper Saddle River, New Jersey.

MURATA, Tadao (1989). Petri Nets: Properties, Analysis and Applications. Proc. IEEE 77(4): 541-580.

PUIGJANER TREPAT, Ramón; SERRANO, J.; RUBIO, A. (1992). Evaluación y Explotación de Sistemas Informáticos. Síntesis. Madrid.

REUTENAUER, Christophe (1990). Mathematics of Petri Nets. Prentice Hall.

¾ The MATH WORKS Inc. Matlab V. 5.3 / Simulink V. 3.0. Licencia No 155062 - CERET - Fac. Regional Santa Fe (Univ. Tecnológica Nacional).