

STATECHARTS ESTOCÁSTICOS PARA EVALUACIÓN DEL DESEMPEÑO EN LOS SISTEMAS REACTIVOS

ALONSO DE JESÚS CHICA LEAL*
DIEGO FERNANDO CANO MENDOZA**

Resumen:

Las cartas de estado (Statecharts) son representadas en un lenguaje gráfico para identificar sistemas reactivos complejos. Se presenta la metodología realizada en [1] donde se realiza una extensión de los diagramas de transición – estado para los cuales las nociones de jerarquía y ortogonalidad han sido introducidas. Recientemente, Statecharts han sido sugeridos para representar modelos de desempeño. En esos modelos de desempeño, el comportamiento de un sistema bajo estudios se considera probabilístico y con este interés se han desarrollado software dedicados a esta tarea. Por consiguiente, la inclusión de probabilidades en protocolo Statecharts es el interés de este artículo. Dado que un sistema reactivo es modelado probabilísticamente para eventos. Este modelo en Statecharts es equivalente una cadena o proceso de Markov en tiempo continuo, por tanto se construye un modelo analítico del modelo y se presenta el aspecto de generación de una cadena de Markov de una representación Statecharts junto con la solución para incluir probabilidades entre las transiciones. En este artículo presentamos los Statecharts a partir de la utilización del Toolbox de Matlab llamado Stateflow.

Palabras clave: Modelos de desempeño; Statecharts; Reacción probabilística; Cadenas de Markov en tiempo continuo.

INTRODUCCIÓN

El alcance y propósito del desempeño de un sistema hoy en día pueden ser analizados antes de llevarlo a la práctica con base en la evaluación de la información. Existen dos estrategias que pueden ser tomadas en consideración con el propósito de conseguir la evaluación del desempeño de un sistema dado. La primera estrategia es la simulación y la otra, la cual es el alcance de este artículo, es la solución analítica. La solución analítica se basa usualmente en el uso de cadenas de Markov de las cuales las probabilidades en estado estable, pueden ser determinadas con base en mediciones de desempeño. La principal preocupación en la obtención de esas mediciones es como precisar los sistemas complejos modernos que reaccionan a los eventos. Un número de esfuerzos en investigaciones están siendo orientados con el propósito de minimizar las especificaciones del problema. La estrategia usada en [1] propone Statecharts, como un método de especificación basada en una representación gráfica, para especificar un sistema complejo reactivo en un alto nivel y luego convertirlo en una cadena de Markov de la cual la medición de desempeño puede ser obtenida. Este artículo discute la necesidad de incorporar transiciones probabilísticas entre estados detectando que un evento puede probabilísticamente moverse a muchos destinos diferentes.

* Nació en León Nicaragua, en 1970. Recibió el grado en Ingeniería Electrónica de la Universidad Autónoma de Colombia en diciembre de 2002. Se encuentra trabajando en la Universidad Autónoma de Colombia como Jefe de área de control y docente de postgrado en la especialización Automática e informática industrial y en el SENA Distrito Capital, centro metalmecánico como diseñador de estructuras curriculares basados en competencias del área electrónica. Es candidato al Título de Magister en Ingeniería Electrónica y de Computadores en la Universidad de los Andes. Los temas de interés son el control híbrido, teoría de control automático, control a prueba de fallas y el control para sistemas no lineales. alonso.chica@fuac.edu.co

** Nació en Bogotá Colombia, en 1983. Recibió el grado en Ingeniería Electrónica de la Universidad de los Andes en diciembre de 2004. Se encuentra trabajando en la Universidad del Bosque como Docente en el área de inferencia estadística del programa de ingeniería industrial. Ha trabajado como docente en la Especialización Automatización de Procesos Industriales de la Universidad de los Andes. Es candidato al Título de Magister en Ingeniería Electrónica y de Computadores en la Universidad de los Andes. Los temas de interés son la identificación on-line de sistemas dinámicos con redes neuronales y control inteligente, para sistemas no lineales. d-cano@uniandes.edu.co

Dos aplicaciones son incluidas: sistema de manufactura y un servidor de archivo de ambiente distribuido, pero otro tipo de sistemas pueden también hacer uso de esta estrategia. La expectativa de este trabajo es posibilitar especificaciones para sistemas complejos en un alto nivel de abstracción y una estructura automatizada para obtener información del desempeño del sistema dado.

MARCO TEÓRICO

Sistemas Reactivos

Son sistemas complejos que interaccionan continuamente con el entorno, recibiendo estímulos del mismo y produciendo salidas en respuesta a los mismos. En otras palabras reacciona a varias clases de eventos, señales, y condiciones. Las siguientes son características de los sistemas reactivos: El orden de los eventos en los sistemas no es predecible, viene determinado externamente. La ejecución de los sistemas reactivos no tiene porque terminar. Los sistemas reactivos son inherentemente no deterministas. Los lenguajes Secuenciales no son adecuados para programar sistemas reactivos. En los sistemas reactivos el orden de ejecución de las instrucciones puede venir determinado por la ocurrencia de eventos internos o externos.

Los sistemas reactivos pueden ser relativamente pequeños, tal como teléfonos celulares, Asistentes personales digitales P.D.A. (Personal Digital Assistant), relojes digitales, sistemas de manufactura, redes distribuidas o muy complejos, tales como los sistemas en automóviles y aplicaciones aeroespaciales. A su vez, éstos pueden ser sistemas de control, tales como controladores telefónicos y de comunicación, robots industriales, o paquetes de software interactivos de gran escala, por ejemplo procesadores de palabra. Los sistemas tales como los sistemas biológicos, el cerebro y sistemas de inmunidad, son vistos como el epítafio de los sistemas complejos reactivos [2].

Statecharts

Son gráficos orientados capaces de modelar sistemas reactivos. Ellos han sido desarrollados para representar y simular sistemas en tiempo real [1]. Ellos son una extensión de los diagramas transición – estado, a los cuales se les adiciona los conceptos de jerarquía (profundidad), ortogonalidad (representación de actividades en paralelo) e interdependencia (difusión de la comunicación). Los elementos básicos que constituye los statecharts son: Estados, Transiciones, Eventos, Condiciones, Acciones, Expresiones, Variables y Etiquetas. [1]

[5] Los estados describen un modo de funcionamiento de un sistema reactivo, éstos pueden ser del tipo “mutuamente” exclusivo (OR) o de tipo paralelo (AND). Los estados tipo OR sólo pueden ser ejecutados uno a la vez, mientras que los te tipo AND pueden accionarse paralelamente (Ver figura 1).

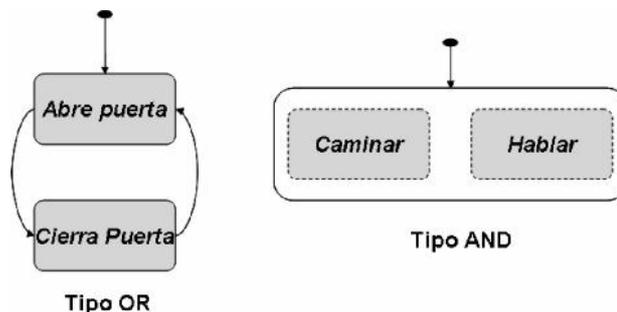


Fig. 1. Estados tipo OR y AND

Las transiciones son los “conectores” de estado a estado; el estado fuente es de donde sale la transición, y el estado destino es donde llega la transición. Las transiciones describen el conjunto de eventos o condiciones que se deben cumplir para pasar de un estado a otro (Ver figura 2). Las transiciones predeterminadas corresponde al conector que tiene una bola en un extremo en vez de estar conectada a un estado; éste tipo de transiciones indican el estado inicial o predeterminado del proceso, es decir, en el momento que inicie a operar el sistema iniciará donde la transición predeterminada le indique.

Las acciones, pueden ser llevadas a cabo dentro de los estados o en las transiciones; éstas pueden corresponder a la activación de una señal, al incremento en algún contador o a la modificación de una variable entre otras.

Dentro de los elementos que poseen los statecharts, se encuentran también los “puntos de decisión” (Connective Junctions). Éstos facilitan la representación de un sistema, especialmente cuando éste requiere la toma de decisiones para una determinada acción. Estos elementos son de vital importancia para poder incluir grados de probabilidad al diagrama.

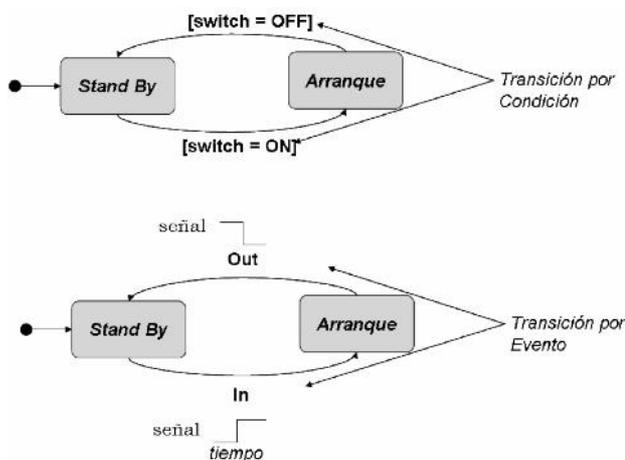


Fig. 2. Tipos de Transiciones

Definición de los statecharts estocásticos

Los statecharts estocásticos parten del supuesto que los eventos que se generan dentro de éstos tienen distribución exponencial, así, un proceso descrito con esta técnica puede ser modelado con una cadena de Markov.

Por otra parte, las probabilidades juegan un papel muy importante en las cadenas de Markov, ya que de acuerdo a éstas, un proceso puede ser evaluado basándose en los parámetros de desempeño encontrados a través de las probabilidades de estado del mismo. De esta forma, se puede extender la capacidad de modelamiento de los statecharts mediante la utilización de probabilidades en éstos.

A continuación se explica brevemente el procedimiento para incluir las probabilidades dentro de un statechart y se ilustra con una figura.

Probabilidades en las transiciones

[1] Las probabilidades pueden ser incluidas en las transiciones de los statecharts cambiando la sintaxis original *evento[condición]/acción* a la sintaxis *evento [condición][probabilidad]/acción*. La modificación se refiere a la asociación de valores de probabilidad para cada posible camino tomado por un evento. Esta situación fuerza a los eventos a optar en tomar un camino entre varias posibilidades. En este caso la transición es disparada con base en las probabilidades provistas para cada uno de los estados destino [1].

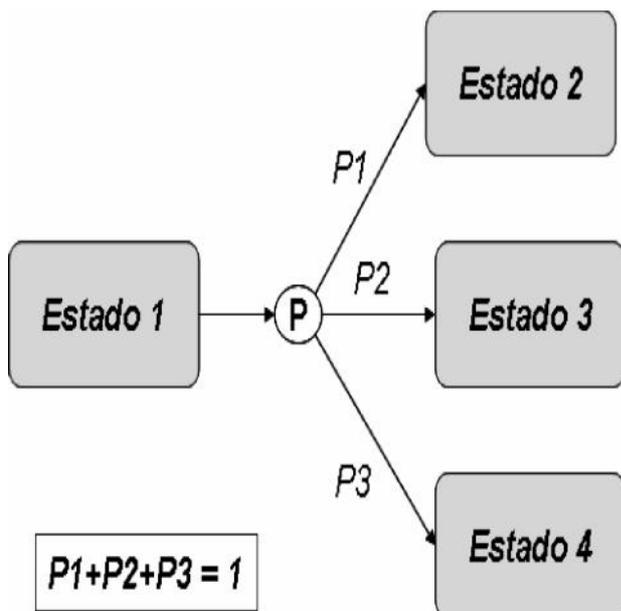


Fig. 3. Punto de decisión condicional

Procesos de Markov

Un proceso de Markov es un proceso estocástico para el cual la ocurrencia de un estado futuro solo depende del estado inmediatamente anterior. Es decir es un proceso sin memoria. Este proceso está caracterizado por estados discretos y los eventos por transiciones entre estados. Las transiciones de estado corresponden a cambios de un estado en otro.

Un proceso de Markov se identifica como sigue: Si $S_i(n)$ identifica al evento de que el sistema está en el estado i después de n transiciones, entonces:

$$P\{S_i(n) / S_a(n-1), S_b(n-2), \dots, S_z(0)\} = P\{S_i(n) / S_a(n-1)\}$$

Donde $P_{ij} = P\{S_j(n) / S_i(n-1)\}$ es la probabilidad condicional de que el sistema cambie a S_j , en el tiempo n , dado que estaba en S_i en el tiempo $n-1$ y $P_i(n) = P\{S_i(n)\}$ es la probabilidad de que el sistema está en el estado i en el tiempo n . En general debe cumplirse que $0 \leq P_{ij} \leq 1$ y:

$$\sum_{\forall j} P_{ij} = 1$$

A la representación de los estados en una cadena y proceso de Markov se le denomina diagrama de transición de estados. Las cadenas de Markov pueden ser discretas o de tiempo continuo. En este artículo solo se tratará las cadenas de Markov en tiempo continuo.

Los procesos de Markov de estados discretos pero continuos en tiempo, son de gran aplicación en estudios de confiabilidad y en nuestro caso en estudios de desempeño de sistemas reactivos. Las probabilidades de estado estable que se pueden obtener del proceso de Markov permite asociarlas a relaciones de estado estable para mediciones de desempeño.

De Statecharts a Cadenas de Markov

[1] Una vez han sido incluidas las probabilidades en el modelo statechart del proceso, éste puede ser traducido a una cadena de Markov mediante el monitoreo global de los estados del sistema y sus respectivas transiciones.

Cada vez que ocurre una transición dentro del statechart, se obtiene una configuración diferente para el proceso, esta configuración representa un estado en la cadena de Markov. Así, para cada transición posible se obtiene un nuevo estado en la red Markoviana. En las figuras 4 y 5 se ilustra este fenómeno.

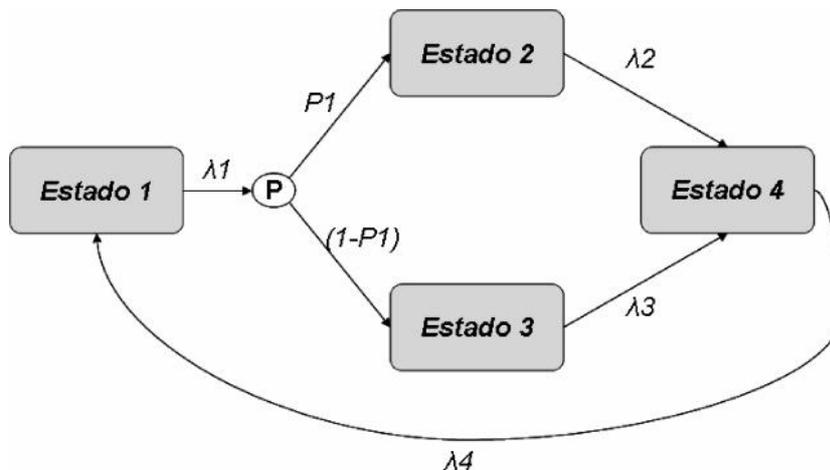


Fig. 4. Proceso de 4 estados modelado por statechart. Las tasas de los eventos que activan las transiciones son λ_1 , λ_2 , λ_3 y λ_4 respectivamente. La probabilidad de ir del estado 1 al estado 2 o 3 es P_1 y $(1-P_1)$ respectivamente.

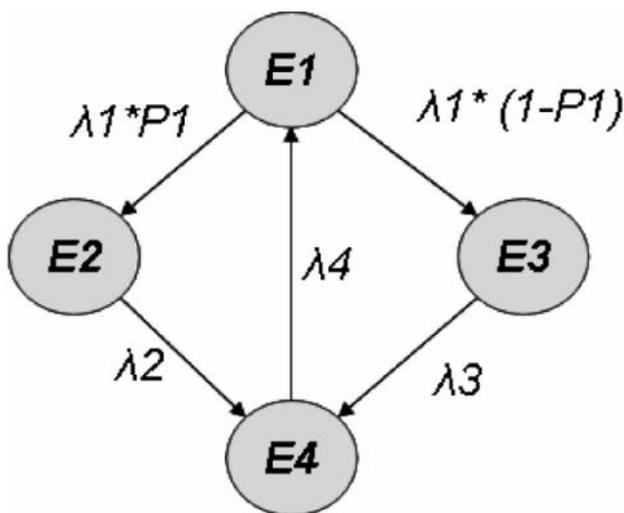


Fig. 5. Cadena de Markov correspondiente al diagrama de la figura 4.

IMPLEMENTACIÓN DE LOS STATECHARTS ESTOCÁSTICOS

El proceso de implementación se puede definir mediante el siguiente algoritmo¹:

1. Definir el proceso, en cuanto a estados y transiciones se refiere.
2. Asignar las condiciones, eventos y probabilidades a las transiciones del modelo.
3. Asignar las acciones a los estados y transiciones que lo requieran.
4. Simular el proceso y extraer los datos de interés (estados y transiciones).
5. Evaluar los parámetros deseados

Cabe anotar que el paso 4 del algoritmo anterior es necesario para la construcción de la cadena de Markov; para la construcción de ésta se requiere tener conocimiento de los estados actual, anterior y la transición que llevó al sistema a cambiar su configuración.

1 NOTA: Este algoritmo pretende sintetizar los pasos de la construcción de un statechart estocástico. No es un protocolo regla para ésta tarea.

# Transición	Estado Actual	Estado Pasado	Tasa
1	E2	E1	$\lambda_1 * P_1$
2	E3	E1	$\lambda_1 * (1 - P_1)$
3	E4	E2	λ_2
4	E4	E3	λ_3
5	E1	E4	λ_4

Tabla 1. Matriz de configuración para la elaboración de la cadena de Markov a partir del statechart. Datos tomados de la figura 4.

Ejemplos de aplicación

Los ejemplos que se muestran a continuación corresponden a un proceso de operación de una máquina de carga/descarga[1] y un servidor de archivos en un ambiente distribuido[6].

Máquina de Carga/descarga

Se tiene una máquina esperando a ser cargada; una vez se carga pasa a realizar un proceso, en éste punto la máquina puede terminar el producto y entrar en modo de descarga, ó entrar en modo de fallos. En el modo de descarga, la máquina espera cierto tiempo para descargar y vuelve al modo de espera; en el modo de falla, un operario soluciona el problema en un tiempo con distribución exponencial también. El modelo se ilustra en la figura 6.

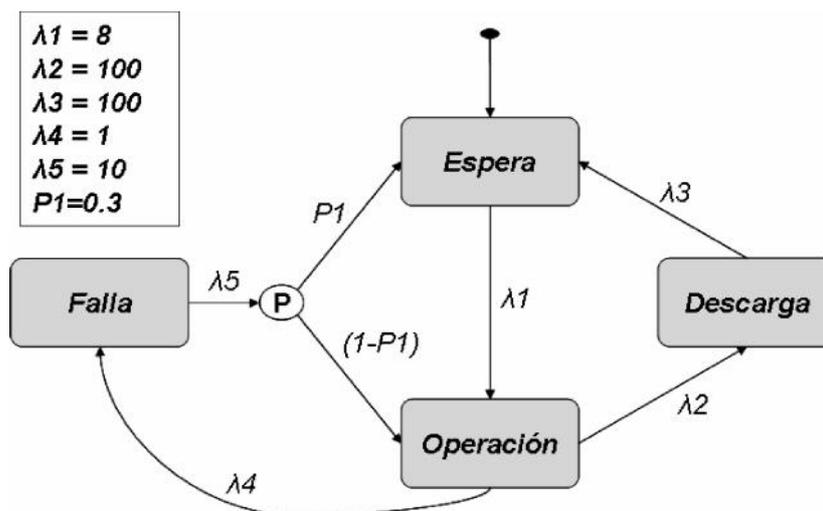


Fig. 6. Statechart de la máquina de carga/descarga..

Para simular el evento estocástico se elaboró una función (universo.m) en MATLAB, que tiene como entrada un vector de n probabilidades y arroja un número que determina la ruta a seguir.

En la figura 7 se muestra la implementación de la máquina con la herramienta stateflow de SIMULINK. Durante la simulación del proceso se tomaron cierta cantidad de datos para la implementación de la cadena de Markov del proceso.

	1	2	3	4
1	-100	100	0	0
2	0	-9	8	1
3	100	0	-100	0
4	7	3	0	-10

Tabla 2. Matriz de transición de estados de Markov para la máquina de carga/descarga.

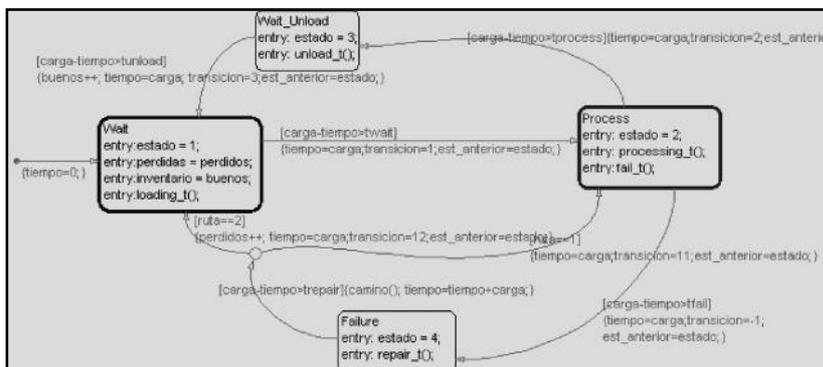


Fig. 7. Implementación en STATEFLOW de la máquina de carga descarga.

A partir de la matriz de transición de estados, podemos evaluar el trabajo de la máquina. En la figura 8 se ilustra la evolución de las probabilidades de estado con el tiempo para cada estado.

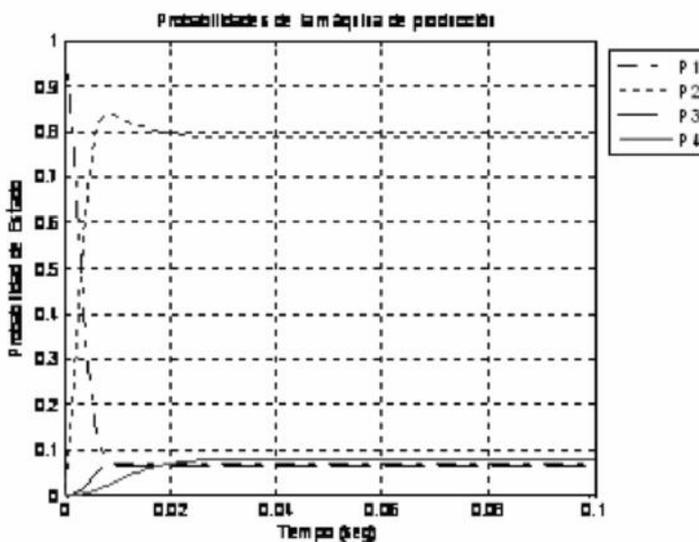


Fig. 8. Probabilidades de Estado para la máquina

Una vez en estado estable las probabilidades determinan la posibilidad de encontrarse en determinado estado. En el caso de la máquina, las probabilidades obtenidas teniendo en cuenta que la probabilidad de se lección de camino P1 es 0.3 fueron:

- o Espera: 0.0657
- o Proceso: 0.7918
- o Descarga: 0.0633
- o Falla: 0.0792

De estos valores se aprecia que la máquina permanece en “proceso” la mayor parte del tiempo.

Servidor de archivos en un ambiente distribuido

[6] El modelo muestra el intercambio entre las consultas de los clientes (Think-Center), una red y un servidor de archivos (modelado por un procesador y un disco). Una consulta hecha por un cliente pasa por la red y regresa por la misma vía. Las tasas son:

- o Clientes = 30
- o Red = 30
- o Procesador = 25
- o Disco = 20.
- o Probabilidad de camino Red = 0.5.
- o Probabilidad de camino Procesador = 0.6

El diagrama de statechart se muestra en la figura 9 y la implementación en stateflow se ilustra en la figura 10.

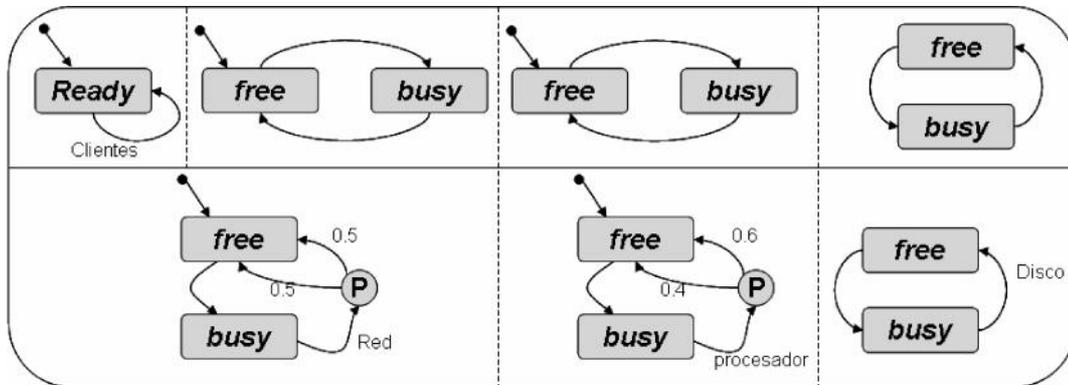


Fig. 9. Diagrama statechart del servidor

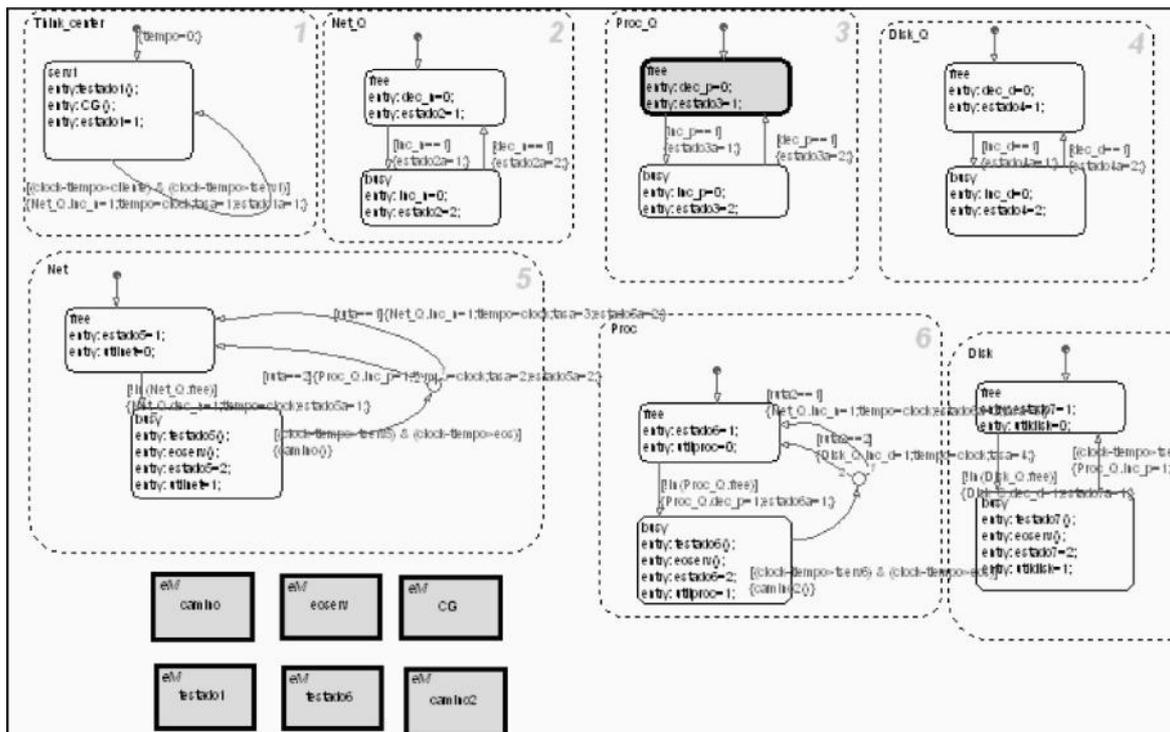


Fig. 10. Implementación del Servidor de Archivos

En las siguientes figuras se muestra la utilización el tiempo de utilización del procesador, la red y el disco.



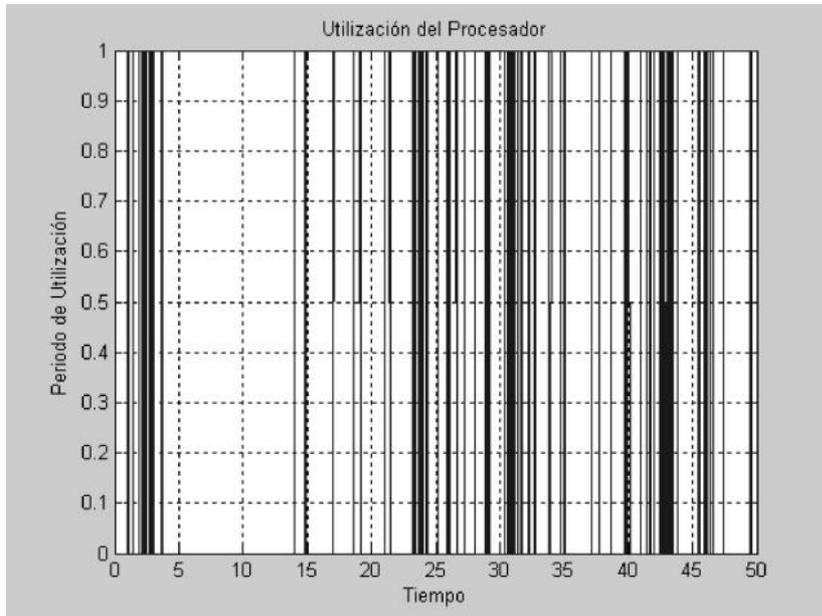


Fig. 11. Utilización del procesador

Según la figura 11, el valor de uno significa que el procesador está ocupado y el de cero que está libre. Así el tiempo promedio de utilización es 39.42 segundos de 50 segundos simulados, lo cual es un promedio de utilización de 78.84%.

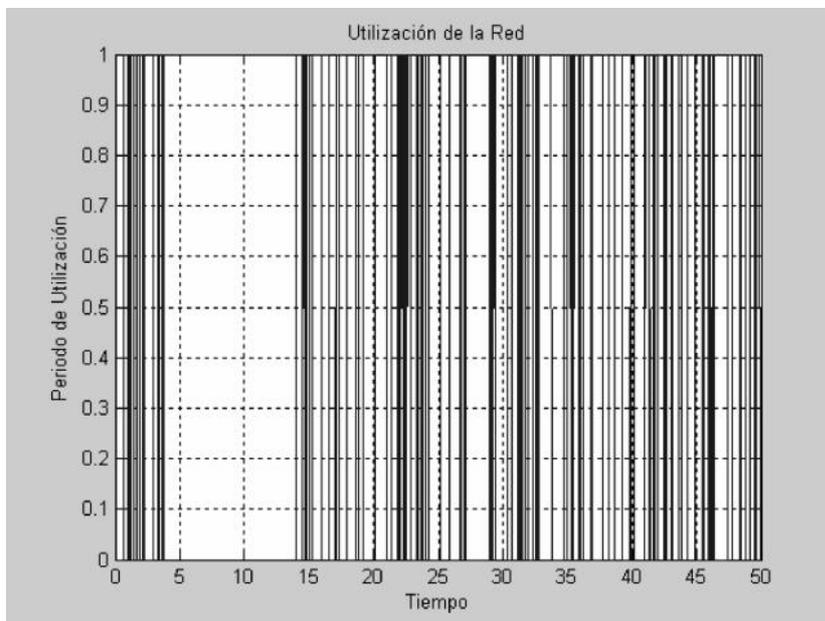


Fig. 12. Utilización de la Red

Para la utilización de la Red según la figura 12, el valor de uno significa que la red está ocupada y el de cero que está libre. Así el tiempo promedio de utilización es 32.44 segundos de 50 segundos simulados, lo cual es un promedio de utilización de 64.88%.

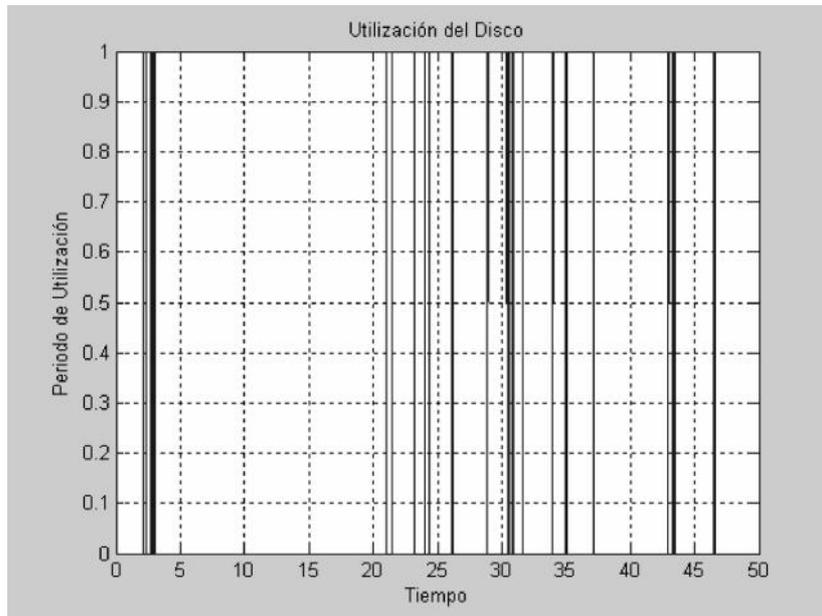


Fig. 13. Utilización del Disco

De acuerdo con la figura 13, el valor de uno significa que el disco está ocupado y el de cero que está libre. Así el tiempo promedio de utilización es 24.53 segundos de 50 segundos simulados, lo cual es un promedio de utilización de 49.06%.

Dada la cantidad de estados, la matriz de transición es extremadamente grande, sin embargo, en la figura 14 se muestran las probabilidades de los estados más significativos del proceso.

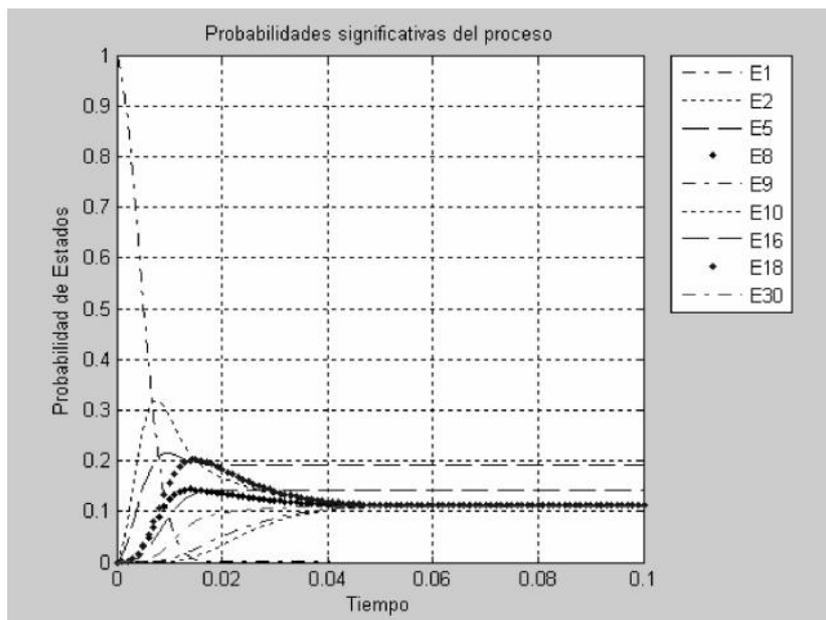


Fig. 14. Utilización del procesador



CONCLUSIONES

Con el uso de las cadenas de Markov se obtuvo una solución analítica, de las cuales las probabilidades en estado estable, se determinaron con base en mediciones de desempeño del sistema.

Para sistemas sencillos las mediciones de desempeño son más fáciles de analizar, en cambio para sistemas más complejos (66 estados) se generan, se requiere un mejor conocimiento de los estados para obtener análisis significativos.

Para obtener el proceso de Markov a partir de los Statecharts es necesario que la tasa de llegada de los eventos describa una distribución de probabilidad exponencial.

Debido a la naturaleza multiestado de diversos sistemas reactivos, la explosión de estados al construir la cadena de Markov es inevitable con el método utilizado. Un posible punto de investigación consistiría en la evaluación de las probabilidades de estado sin la necesidad de construir la matriz de transición.

REFERENCIAS

- [1] N. L. Vijaykumara, S.V. Carvalho, V. M. B. Andradeb, V. Abdurahimanc. Introducing probabilities in Statecharts to specify reactive systems for performance analysis. Science Direct. Abril 2006.
- [2] S. Efroni, D.Harel, I. Cohen, Reactive Animation: Realistic Modeling of Complex Dynamic Systems. IEEE Computer Society. 2005.
- [3] J. Cortadella, A. Kondratyev, L. Lavagno, C. Passerone, Y. Watanabe. Quasi-Static Scheduling of Independent Tasks for Reactive Systems. IEEE transaction on computer-aided design of integrated circuits and systems. Vol. 24, No. 10 October 2005.
- [4] G. Bucci, L. Sassoli, E. Vicario. Correctness Verification and Performance Analysis of Real – Time Systems Using Stochastic Preemptive Time Petri Nets. IEEE transaction on software engineering, Vol. 31, No. 11 November 2005.
- [5] Stateflow and Stateflow Coder. For use with SIMULINK. www.mathworks.com
- [6] C. Lisboa et al. Performance evaluation based on statecharts specification using markov chains and simpl simulation: A case study of a distributed environment with a server file. IEEE transaction on computer-aided design of integrated circuits and systems. June 2001.