

TRABAJO PRÁCTICO N° 2

Herramientas de modelización para sistemas en tiempo real

Objetivos

- Modelar las vinculaciones entre el sistema y el medio utilizando diagramas de contexto.
- Reconocer los eventos externos que influyen un sistema y modelarlos utilizando listas de eventos.
- Representar la evolución de un sistema en el tiempo utilizando diagramas de transición de estados y redes de Petri.
- Conocer y utilizar un software para modelado con Redes de Petri en la simulación de problemas sencillos.

Bibliografía

- *Real-Time Software Design for Embedded System.*
Gomaa Hassan - Cambridge University Press - Capitulo 4

Fecha de presentación: 27/09/2021

Consignas de trabajo

- Repasar los temas vistos en clases de teoría consultando la bibliografía, disponible en el aula virtual de la cátedra.
- Resolver las actividades planteadas a continuación.

Actividades

1. Considerando la siguiente *lista de eventos* para un **sistema de control de trenes**:
 - a. Completar la columna “*Clasificación*”, fundamentando el tipo de evento asignado según los criterios de clasificación vistos en clase.
 - b. Construir el *diagrama de contexto*

EVENTO	RESPUESTAS	CLASIFICACIÓN
On del sistema	On semáforos entradas On semáforos salida On sensores On tablero Monitorear via A Monitorear via B Evaluar horarios	
Señal de presencia en entrada a estación y estación vacía	Poner en verde semáforo de entrada Estación=ocupado Desplegar mensaje	
Señal de presencia en entrada a estación y estación vacía	Poner en verde semáforo de entrada Estación=ocupado Desplegar mensaje	
Operador informa problema	Poner rojo al semáforo de salida. Desplegar mensaje	
Tiempo de horario de salida cumplido	Poner en verde semáforo de salida de estación Estación =libre Desplegar mensaje	
Activación realizada	Poner en verde semáforo de salida Desplegar mensaje	

2. Para el problema del mundo real descrito en el último ejercicio del Trabajo Práctico N° 1:
 - a. Representar las relaciones del sistema con el medio utilizando un **diagrama de contexto**.
 - b. Elaborar la **lista de eventos**.
3. Para cada uno de los sistemas detallados a continuación, construir:
 - a. La **lista de eventos**
 - b. El **diagrama de transición de estados**
 - c. La **red de Petri interpretada**

3.1.a. Objetivo del sistema

Controlar el funcionamiento de un brazo robótico que realiza cirugías a distancia.

3.1.b. Descripción del sistema:

1. El robot es una columna de 1,80 m. con tres brazos. Dos se corresponden a los brazos del cirujano y manejan los instrumentos, y el tercero tiene una cámara de video que filma la operación y la retransmite a un visor para que sea visualizada por el cirujano. El robot ejecuta las órdenes del cirujano.
2. El cirujano ubicado a distancia simula efectuar la operación con un guante con sensores. Estos sensores detectan el movimiento de los dedos del cirujano y lo informa al sistema para que éstos sean validados y transmitidos al robot.
3. Todos los movimientos del cirujano están definidos y registrados en una base de datos propia. Entre estos movimientos se incluye la petición de cambio de instrumento.
4. Un médico asistente ubicado a la par del robot, cambia de instrumentos del brazo del mismo, el que posee un sensor que detecta la presencia del instrumento.
5. El cirujano visualiza toda la operación en un visor tridimensional.

3.1.c. Requisitos funcionales del sistema:

1. Inicializar el robot a partir de la activación del sistema por parte del cirujano.
2. Interpretar (traducir) los movimientos del cirujano y transmitirlos al robot.
3. Visualizar la imagen que captura la cámara del robot, y ubicarla en el visor del cirujano.
4. Informar al asistente, mediante una alarma sonora, cuando se debe cambiar de instrumento.
5. En caso de producirse una falla como la no disponibilidad del instrumento solicitado en el brazo del robot (falla del sensor correspondiente), se emite un mensaje en el visor del cirujano. El cirujano avisa entonces al ayudante quienes continúan con la operación manualmente y el robot queda en espera hasta que el sistema corrige la falla activando una luz verde de "brazo en condiciones".

3.1.d. Requisitos temporales

1. Cumplidos los 30 segundos de la activación de la alarma sonora, ésta se desactivará.

3.2.a. Objetivo del sistema

Controlar la atmósfera de un invernadero.

3.2.b. Descripción del sistema

Para el florecimiento de una orquídea, la temperatura y la humedad del invernadero deben regularse para asegurar que no traspase ciertos límites. La temperatura será medida por medio de un simple termómetro, que proporciona una lectura constante de la misma.

La humedad puede medirse por medio de un simple hidrómetro y será regulada por medio de aberturas en el techo del invernadero. Las mismas pueden ajustarse a cualquier ángulo entre 0 (completamente cerrada) y 180 (completamente abierta).

3.2.c. Requisitos funcionales

1. Todo el sistema podrá ser activado o desactivado por un operador usando un simple control.
 2. Los parámetros Temperatura y Humedad son ingresados por el operador una vez activado el sistema.
 3. La temperatura se regulará usando un calentador, el cual posee un elemento de calentamiento (que sólo posee un ajuste de calor - es decir on/off) y un ventilador que puede ser controlado independientemente. Normalmente, para calentar el invernadero, el calentador y el ventilador se colocan en on. No obstante, si el invernadero comienza a estar demasiado caliente, el ventilador puede mantenerse encendido sin el elemento de calentamiento, con lo cual comenzará a enfriarse.
 4. La relación entre el ángulo de abertura y el grado de humedad se obtiene desde un archivo que es entrada al sistema. Si la humedad del invernadero está fuera del rango, las aberturas serán ajustadas apropiadamente.
 5. Como toma algún tiempo para que los niveles de humedad bajen después de que las aberturas han sido ajustadas, no se realiza otra lectura de humedad hasta pasados los 5 minutos, para permitir que el cambio de posición de las aberturas tenga efecto.
4. Evalúe el cumplimiento de las propiedades de la red de Petri para los sistemas del ejercicio 3, aplicando análisis por enumeración.
 5. Simular las redes Petri desarrolladas, utilizando el Software *Visual Object Net++*.