Statecharts Parametrizados

Maximiliano Cristiá
Análisis de Sistemas
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario

Abril de 2012

Resumen

En este apunte se presenta un ejemplo en el cual se hace un uso intensivo de Statecharts parametrizados y además se explican y clarifican ciertos aspectos de estos Statecharts que David Harel sugiere en su trabajo seminal. Sólo se presenta una parte del conocimiento del dominio y la especificación del problema, el resto de las descripciones de WRSPM quedan como ejercicio para el alumno.

1. Regla fundamental para el uso de Statecharts parametrizados

Los Statecharts parametrizados son una forma de expresar Statecharts complejos, esto significa que no hay una semántica específica para ellos sino que se debe usar la semántica de un Statechart no parametrizado. En consecuencia la regla para el uso de Statecharts parametrizados es la siguiente: un Statechart parametrizado es correcto si puede ser traducido automáticamente (mecánicamente, es decir con un algoritmo) a un Statechart no parametrizado. La posibilidad de traducción automática está directamente ligada a que la traducción no implique cuestiones semánticas sino únicamente sintácticas.

2. Requerimientos informales

Una habitación posee varias puertas de ingreso/egreso. En cada puerta se han instalado dos sensores ópticos uno al lado del otro (cada uno es semejante a los que se instalan en las puertas de los ascensores para evitar que se cierren si hay alguien entrando o saliendo). Estos sensores envían una señal cada vez que algo interrumpe el haz de luz que se establece entre cada extremo. Como hay dos por puerta es posible determinar si una persona ingresa a la habitación o sale de ella. Notar que esta inteligencia NO es parte de la funcionalidad de los sensores.

La habitación cuenta con un sistema de ventilación electrónico. Es posible aumentar o disminuir discretamente la cantidad de aire que el sistema hace circular.

Se requiere un programa que aumente/disminuya paulatinamente la circulación de aire desde cero hasta el máximo posible, cada vez que ingresan/egresan tres personas a la sala. Es decir la potencia de circulación debe ser la misma, por ejemplo, si hay 3, 4 o 5 personas en la sala.

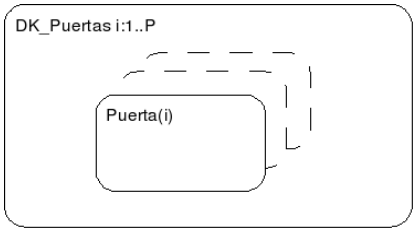

Figura 1: P puertas en paralelo usando la sugerencia de Harel.

3. Desginaciones

Como en este apunte solo modelaremos parte del conocimiento de dominio (DK) y la especificación (S) daremos únicamente las designaciones necesarias para describir esos documentos.

Se aumenta en una unidad la potencia del sistema de ventilación $\approx A$

Se disminuye en una unidad la potencia del sistema de ventilación $\approx D$

El sensor i de la puerta p emite una señal $\approx s(i, p)$

La cantidad de puertas de la habitación $\approx P$

La potencia máxima del sistema de ventilación $\approx PM$

4. El conocimiento del dominio (DK)

Como mencionamos más arriba sólo modelaremos una parte del conocimiento del dominio; el resto queda como ejercicio para el alumno. Comenzamos modelando P puertas en paralelo. Para esto usamos un Statechart parametrizado paralelo (o de tipo AND) como lo sugiere Harel [1], aunque usamos un grafismo ligeramente diferente al propuesto por el autor, ver Figura 1.

Notar que también hemos modificado la forma de nombrar los estados parametrizados, con $Puerta(i)$, en lugar de con $Puerta\ i$.

Seguimos en la Figura 2 con el modelo de cada puerta la cual consiste en dos sensores que emiten una señal cada vez que algo los activa. Ambos sensores están en paralelo. Sin embargo, se espera que haya cierta serialización en las señales: si $Sens1$ se activa entonces luego se activa $Sens2$ y viceversa; dos activaciones consecutivas de uno de los sensores significa que la persona se arrepintió de entrar o salir, pero es imposible que dos personas activen una única vez cualquiera de los sensores ni que se active dos veces consecutivas uno de ellos sin que en el medio se active el otro. Para modelar todo esto o bien hay que modificar el modelo actual o bien agregar más conocimiento de dominio que indique que se está asumiendo tal comportamiento. Sin embargo, para mantener el problema simple no modelarmos ninguna de estas propiedades del domino del problema (el resto lo dejamos como ejercicio para el lector).

Notar que se utilizan los eventos designados para modelar las señales de los sensores.

Ahora pasamos al conocimiento de dominio referente al sistema de ventilación. Aquí utilizamos un Statechart parametrizado de tipo OR como se muestra en la Figura 3. Harel, en lugar de elipses, utiliza rombos. El conjunto de estados parametrizados ($Pot(j)$) comprende los estados $Pot(j)$ con j entre 1 y $MP - 1$, lo que implica que ni $Pot(0)$ ni $Pot(MP)$ están en ese conjunto.

Una flecha saliendo del conjunto de estados parametrizados ($Pot(j)$) hacia un estado elipse significa que de cada estado (j) del conjunto sale una flecha idéntica al estado cuyo índice es el indicado en el elipse. En nuestro ejemplo, tenemos que de cada $Pot(j)$ sale una flecha etiquetada con A a

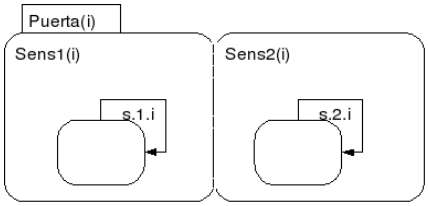

Figura 2: Un modelo simple para el DK de los sensores de cada puerta.

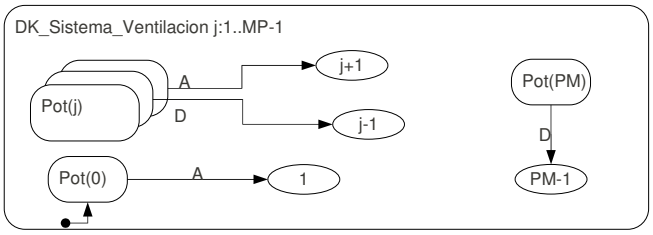

Figura 3: El conocimiento de dominio sobre el sistema de ventilación.

$Pot(j+1)$ y una etiquetada con D a $Pot(j-1)$. Notar que si j vale 1 hay una flecha etiquetada con D a $Pot(0)$ aunque este último no es parte del conjunto; algo similar vale para $Pot(MP)$. Es responsabiliad del especificador que tales estados ($Pot(0)$ y $Pot(MP)$) existan en el Statechart. También, aunque no aparece en el ejemplo, una flecha saliente del conjunto de estados parametrizados hacia un estado común tiene el mismo significado: de cada estado parametrizado sale una flecha hacia ese estado con la misma etiqueta. Aunque tampoco aparece en este ejemplo, es posible tener flechas que entren al conjunto de estados parametrizados desde estados no parametrizados (comunes); en este caso el significado es similar: una flecha desde el estado común hacia cada uno de los estados parametrizados con la misma etiqueta (lo que es no determinístico).

No tiene mucho sentido tener flechas entrantes al conjunto provenientes desde un elipse.

5. La especificación (S)

Hemos especificado el sistema como dos máquinas en paralelo como se muestra en la Figura 4.

Comenzamos expandiendo *PuertasInteligentes* (Figuras 5 y 7) que, precisamente, hace lo que los sensores de cada puerta no pueden hacer por sí sólos: determinar cuándo una persona entra y cuándo sale. Para ellos pensamos en un "proceso" por puerta que atiende las señales de sus sensores y determina si por esa puerta sale o entra una persona. Las salidas o entradas se comunican al resto del sistema por medio de los eventos s y e, respectivamente. Notar que en *PuertasInteligentes* el índice es k mientras que en $DK\_Puertas$ es i. Esto no causa problemas pues como indica la regla para la construcción de Statecharts parametrizados, luego de la traducción de ambos a Statecharts no parametrizados los índices desaparecen siendo reemplazados por constantes numéricas, por lo que,

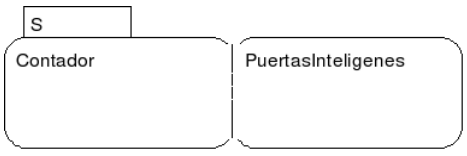

Figura 4: La especificación al más alto nivel de abstracción.

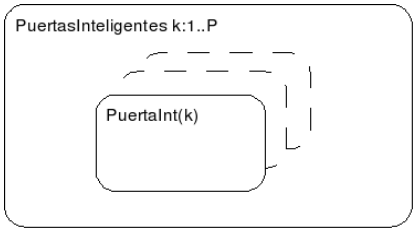

Figura 5: *PuertasInteligentes* es simplemente la composición en paralelo de cada puerta.

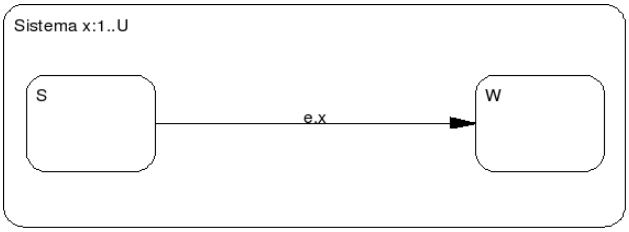

Figura 6: Parametrización de transiciones.

por ejemplo, los eventos de la forma $s.1.i$ y $s.1.k$ (de *DK_Puertas* y *PuertasInteligentes*) pasan a ser los mismos: $s.1.1$, $s.1.2$, etc.

En la Figura 8 mostramos la parte del sistema que cuenta las entradas y salidas según lo indicado por cada *PuertaInt* y comunica al sistema de ventilación si debe disminuir o aumentar la potencia cada tres salidas o entradas, respectivamente. En este caso hemos denotado el estado inicial pintando de gris uno de los estados; lo hicimos de esta forma para aprovechar los cuatro puntos de conexión del rectángulo para conectar transiciones de estado.

6. Un problema con la especificación

Si entra una cantidad suficiente de gente el sistema de ventilación llegará a su máxima potencia. Si continúa entrando gente, el sistema de software continuará enviando señales al sistema de ventilación para que este siga aumentando la potencia pero serán ignoradas. Ahora, si comienza a salir gente, el sistema de software pedirá al sistema de ventilación que reduzca la potencia cuando en realidad no debería hacerlo pues aun hay demasiada gente dentro de la habitación.

7. Parametrización de transiciones

Una forma de parametrización que no apareció en el ejemplo de este apunte se da cuando es necesario que una máquina transicione debido a un gran número de eventos. Graficamos una situación así en la Figura 6. Ese Statechart parametrizado es equivalente a un Statechart no parametrizado en el cual salen U flechas desde el estado S hacia el estado W cada una de las cuales está etiquetada por un evento de la forma $e.i$ para $i \in [1, U]$.

Notar que esta forma de parametrización se puede combinar con las otras.

8. Nota final

Los problemas como el mostrado en este ejemplo llevan a los Statecharts a sus límites expresivos. Es posible que sea conveniente utilizar lenguajes como CSP o TLA, o combinar Statecharts con Z

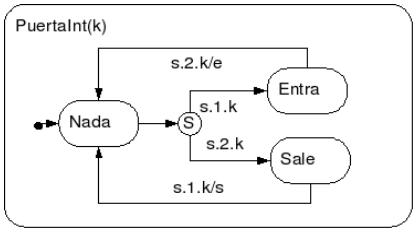

Figura 7: El modelo para cada puerta determina si una persona entra o sale según el orden de las señales provenientes de sus sensores.

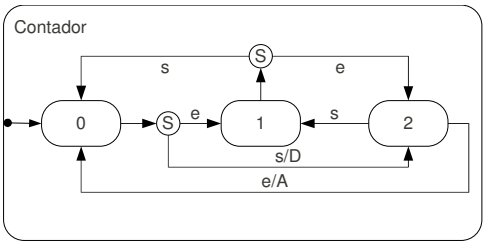

Figura 8: Contador de personas que entran o salen por cualquier puerta.

para este tipo de problemas.

Ejercicio 1 *Modificar el conocimiento de dominio respecto de los sensores de cada puerta de forma tal que se cumplan las siguientes propieades:*

- *Dos activaciones consecutivas de uno de los sensores significa que la persona se arrepintió de entrar o salir.*
- *Es imposible que se active dos veces consecutivas uno, sin que en el medio se active el otro.*
- *El primero en activarse siempre será Sens*1*.*

Ejercicio 2 *Busque una solución para el problema planteado en la sección 6.*

Referencias

[1] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, pp. 231–274, 1987.