Unidad 6. Modelado de estados

En la unidad anterior aprendimos que los modelos dinámicos son imprescindibles para verificar que un modelo estático es correcto. Debido a la importacia de estos modelos en UML no únicamente existen los diagramas de interacciones, también están los diagramas de estado que, al igual, representan un modelado dinámico. Se tienen los diagramas de estado que están orientados a modelar los cambios de estado de un objeto, lo anterior, usando el concepto de Maquinas de Estado Finito.

¿Qué es una máquina de estado finito?

Considere el acceso a la entrada del metro, dicho acceso esun sistema que permite la entrada de una persona ya que ha introducido una moneda. El funcionamiento de este dispositivo se puede ver como una máquina de estado finito (MEF). La figura 28 muestra una parte de la MEF. Los óvalos son los estados, el mecanismo solo tiene dos. Puede estar bloqueado o desbloqueado. Cuando el mecanismo esta bloqueado, una persona puede introducir una moneda, para que el mecanismo quede desbloqueado. Lo anterior se muestra en el diagrama por una flecha que une los estados bloqueados y desbloqueados. La flecha es llamada transición porque describe como son las transiciones en la MEF.

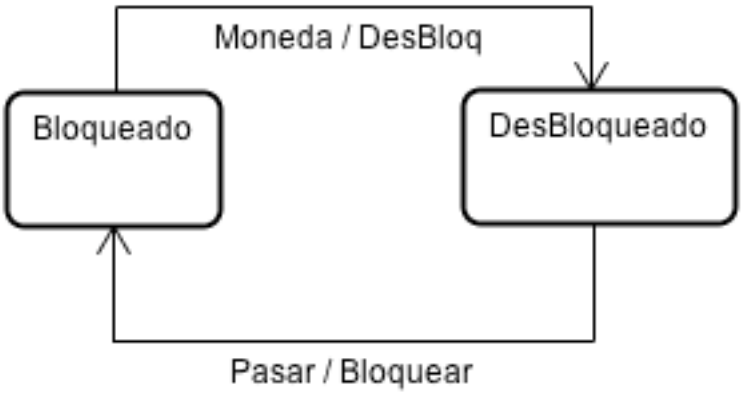


Figura 28. MEF del mecanismo para acceso al metro.

La etiqueta en la transición tiene dos partes separadas por una línea. La primera es el nombre del evento que dispara la transición. La segunda es el nombre de la acción a realizarse cuando la transición ha sido disparada. La figura 28 puede interpretarse de la siguiente manera.

- Si el mecanismo está en estado bloqueado, y el evento “Moneda” ocurre, entonces el mecanismo pasa al estado desbloqueado y la acción Desbloq es ejecutada.
- Si el mecanismo está en estado desbloqueado, y el evento “Pasar” ocurre, entonces el mecanismo pasa al estado bloqueado y la acción Bloquear es ejecutada.

Lo anterior describe como el mecanismo trabaja cuando las cosas ocurren como son planeadas. Asumamos que el estado inicial es bloqueado. Cuando una persona quiere pasar por el mecanismo, debe de depositar una moneda. Lo anterior, dispara el evento “Moneda”, el cual, cuando el mecanismo está en estado bloqueado causa que se de una transición al estado desbloqueado y la ejecución de la acción Desbloq. A continuación la persona para por el mecanismo, lo cual , dispara el evento “Pasar” que causa una transición al estado bloqueado e invocar a la acción Bloquear.

Lógica Anormal

Cláramente, el diagrama de la figura 28 es una forma elegante para describir la lógica del mecanismo en cuestión. La presentación visual de la lógica es un aspecto poderoso de los diagramas de estado. Lo que lo hace aún más poderoso es lo fácil que podemos saber que el diseño está finalizado.

Hay que notar que en nuestro ejemplo es posible adicionar eventos a los estados que ya tenemos. Ver figura 29.

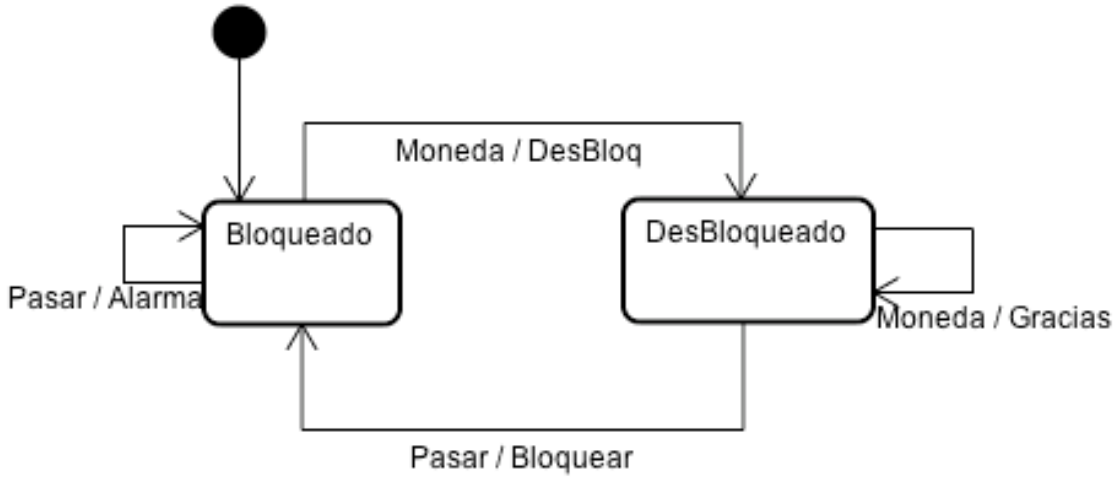


Figura 29. Mecanismo con eventos anormales.

¿Qué haríamos si el mecanismo esta en estado bloqueado, pero el usuario pasa?. Es claro que una alarma debe ser activada, esto se puede ver en el diagrama anterior. Notese que el mecanismo no cambia de estado y permanece como bloqueado. La otra condición anormal es cuando el mecanismo está desbloqueado y el cliente deposita otra moneda. En esta caso se prenderá la luz de “Gracias”.

Condiciones anormales como las anteriores ocurren cuando eventos normales pasan inesperádamente. La perversidad del mundo nos garantiza que esas condiciones ocurrirán, quizás más frecuentemente de lo que podríamos pensar, lo anterior en condiciones no controladas.. Es fácil subestimar esos escenarios y caer en serios problemas. Las MEF son una buena forma de descubrir esas condiciones en etapas tempranas del desarrollo y ver que ocurre cuando esas situaciones anormales se presentan.

En la figura 29, existe un círculo negro con una flecha apuntando al estado bloqueado. El círculo es llamado pseudo estado inicial, esto significa que el estado bloqueado es el

inicial para la MEF. Cuando el MEF es activado por primera vez, comenzará con el estado bloqueado.

Una buena forma de manejar alarmas.

Recordando que el estado bloqueado, no es el mejor para lidear con la situación anormal de que el usuario quiere forzar su entrada. Debemos pasar a un estado llamado, quizás, violación quede ahí hasta que el mecanismo sea reparado para volver a dar servicio. Lo anterior se puede ver en la figura 30.

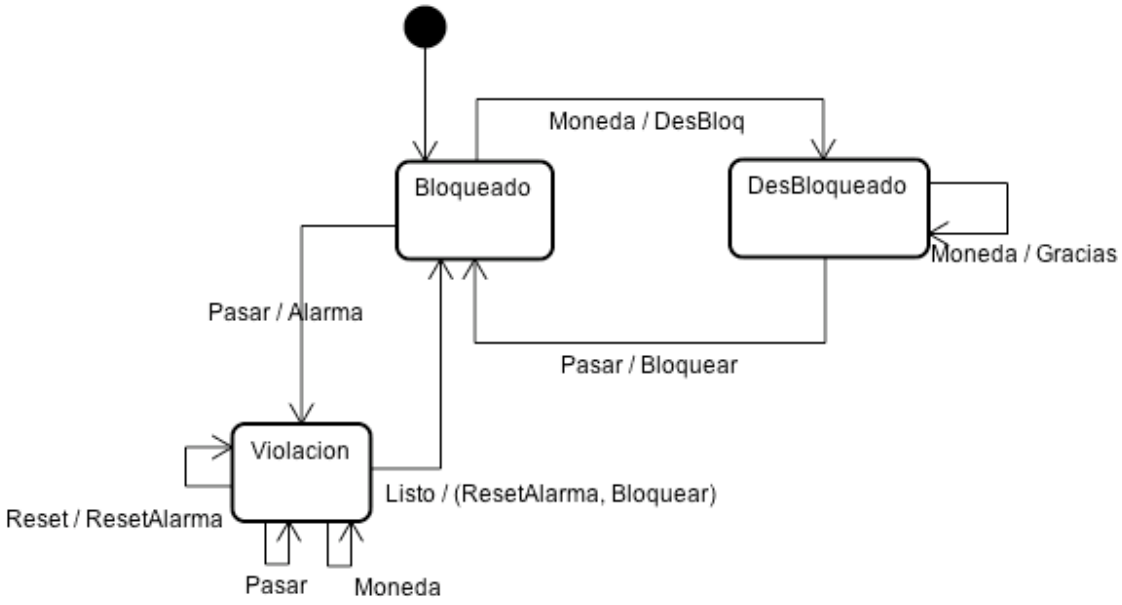


Figura 30. Mecanismo de acceso a metro con estado Violacion

Nótese que la única manera de salir del estado Violacion es con el evento Listo. La transición asegura que la alarma es apagada y el mecanismo es bloqueado de nuevo. Hemos adicionado un evento que el técnico puede usar apagando la alarma mientras trabaja en el mecanismo. Se puede ver que los estados Pasar y Moneda son totalmente ignorados.

También hemos añadido una acción a la flecha que une el pseudo estado inicial con el estado bloqueado. Esta línea es la primera transición que ocurre cuando la MEF es iniciada. La acción garantiza que el mecanismo estará bloqueado al principio de su operación.

Hay que notar que no se añadió el evento listo a los estados bloqueado y desbloqueado, ya que no es manejado por estos, y sería un error si ocurriera. Como un comportamiento por defecto, se asume que en el MEF todo evento que ocurra en un estado no permitido se manejará como un error fatal.

Modo Diagnóstico

Es muy deseable que exista un modo de diagnóstico para que los técnicos realicen pruebas o den mantenimiento al dispositivo. Es dicho modo el técnico podrá realizar comandos a varias funciones del dispositivo y probar sus sensores. La figura 31 muestra el posible diseño.

Ciertamente las cosas se han tornado un poco más complicadas debido al modo de diagnóstico. Una de las más cosas más llamativas del diagrama son los rectángulos que envuelven a los estados. Estos rectángulos son llamados super estados. Existen dos super estados en nuestro sistema, el normal y el de diagnóstico. Los estados se pueden identificar como super y sub estados.

Como se puede observar, los estados y transiciones para la operación normal del mecanismo están contenidos en el modo normal. Nada cambió en dicho estado, a excepción de algunas acciones de apagado de la luz indicadora “Gracias” cuando el usuario pasa por el mecanismo.

Para pasar al modo de diagnóstico, los técnicos deben de iniciar el evento Diagnóstico, es importante observar que la línea para este evento parte del super estado y no de ninguno de los sub estados que se tienen en el modo normal, lo anterior indica que, no importando en que estado se encuentre el mecanismo al momento de estar operando, es posible ponerlo en modo de diagnóstico. Al pasar al modo de diagnóstico se ejecuta la acción salvarEstadosDisp que simplemente almacena el estado en que se encuentra el dispositivo antes de pasar a diagnóstico.

Superestados son como clases abstractas, pueden ser instanciados usando una derivado. Cuando se entra al modo de diagnóstico se debe pasar a uno de los sub estados, en cual?, en el que es referido como estado inicial, probar moneda.

El técnico puede probar el sensor de recepción de monedas, así como el funcionamiento de la luz indicadora de “Gracias”, lo anterior usando los estados probar moneda y probar pasar. En cualquier momento el técnico puede probar la alarma y el bloqueo del mecanismo, dichas pruebas no alterarán el estado en el que se encuentre.

Habiendo terminado las pruebas el técnico puede retornar el mecanismo al estado normal de dos formas. Utilizando el evento inicializar, que pone al dispositivo en estado bloqueado con la alarma apagada así como la luz indicadora, o utilizando el evento retornar, este regresa el dispositivo al estado en que estaba cuando entró al modo diagnóstico. El pseudo estado Historial es un pequeño círculo con una H dentro. Este indica que el subestado normal que se tiene es el mismo que se tenía cuando se abandonó el modo normal.

Implementando las maquinas de estado finito.

Hay varias formas de implementar las máquinas de estado finito. Una de las más comunes es el uso del switch y case. A continuación se muestra la implementación del diagrama de la figura 29.

```
 int bloqueado=1;int desbloqueado=2;

int pasar = 10; int moneda = 11;
void desbloq(){}
void bloq(){}
```

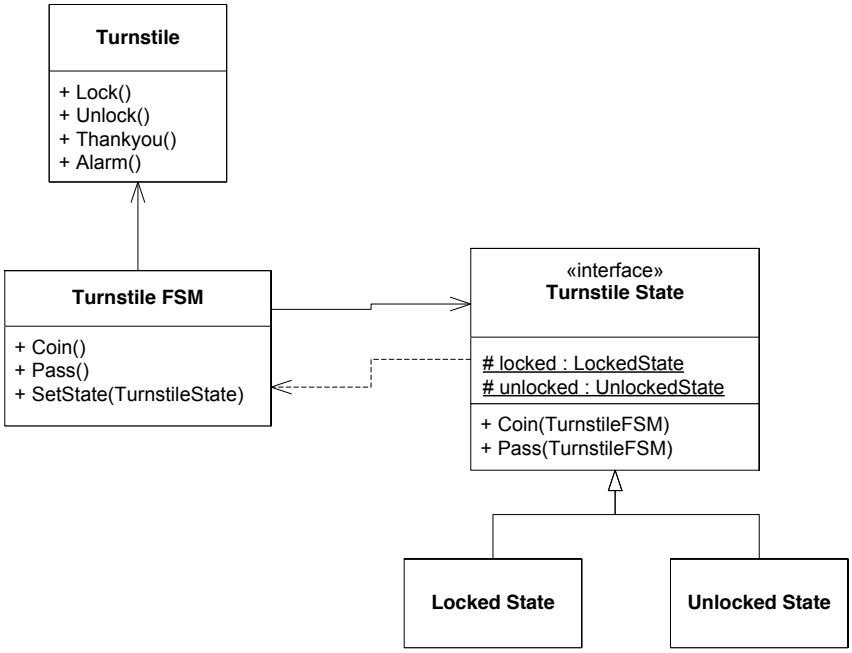
```
void gracias(){}
void alarma(){}
static  int estado = bloqueado;
void Transition(int e)
{
   switch(estado)
   {
   case bloqueado:
      switch(e)
      {
      case moneda:
         estado = desbloqueado;
         desbloq();
      break;
      case pasar:
         alarma();
      break;
      }
   break;
   case: desbloqueado
      switch(e)
      {
      case moneda:
         gracias();
      break;
      case pasar:
         estado = bloqueado;
         bloquear();
      break;
      }
   break;
   }
}
```

A pesar de que la técnica es común no es muy bonita ni elegante, de hecho se complica mucho cuando empieza a complicarse los estados. Se pueden tener página tras página de código muy similar para manejar los distintos estados.

Usando el patrón state

Es posible emplear el patrón state. En la figura 32 es posible observar como es aplicado el patrón state a nuestro problema. Comenzamos con la clase Dispositivo que implementa las cuatro acciones: bloquear(), desbloquear(), alarma() y gracias(). Derivada de esta se tiene la clase DispositivoFSM que implementa los eventos Moneda y Pasar. DispositivoFSM contiene un apuntador a la interface DispositivoState. Las funciones de los eventos de la clase DispositivoFSM se delegan a su contraparte en DispositivoState. Exisen dos clases derivadas de DispositivoState: EstadoBloq y EstadoDesbloq. Instancias de estas clases accesan variables estáticas de DispositivoState.

Si la MEF está en estado bloqueado, el DispositivoFSM apuntará al objeto de tipo EstadoBloq. Si ocurre un evento Moneda esta es trasladado a un objeto de tipo DispositivoState que en este caso es EstadoBloq por lo que, debido al polimorfismo la llamada a la función Moneda es resuelta por un objeto de tipo EstadoBloq.



Ejercicios

- **Línea Telefónica:** Se requiere modelar el funcionamiento de una línea telefónica cuando se realiza una llamada, la recepción de llamadas no debe estar dentro del modelado. La secuencia de pasos que se realizan son los siguientes:

  1. Se descuelga el teléfono (usuario)
  2. Provee un tono de llamado (teléfono) Ex 2.1
  3. Se ingresa el número telefónico (usuario) Ex 3.1, Ex 3.2
  4. Teléfono valida número y realiza el enrutamiento de la llamada.
  5. Teléfono provee tono de llamada
  6. Se realiza la conexión cuando se contesta la llamada
  7. El teléfono al que se llamó termina la llamada.
  8. Teléfono se desconecta

  Ex 2.1. (8)

  1. Se acaba el tiempo y no se ha ingresado ningún número.
  2. Teléfono manda tono de advertencia.
  3. Teléfono pasa al estado de tiempo terminado.

  Ex. 3.1

  1. Se ha terminado el tiempo para ingresar los números
  2. Teléfono manda un tono de advertencia

3. Teléfono pasa al estado de tiempo terminado.

Ex. 3.2

1. Se ha marcado un número incorrecto
2. El teléfono manda mensaje.
3. El teléfono se desconecta.

Nota: En cualquier momento el usuario puede colgar el teléfono.

- **Copiadora:** Se requiere el modelado del funcionamiento de una copiadora simple. La secuencia de actividades es la siguiente.

  1 Se prende la copiadora.

  1.1 Copiadora entra en estado de espera.

  1.2 Se realiza pruebas automáticas para verificar el estado de la impresora

  1.3 Parpadea el indicador de "listo"

  1.4 Termina las pruebas

  2 Copiadora lista para comenzar.

  3 Usuario indica el número de copias a realizarse (con los botones de incrementar o decrementar) FA 3.1, F.A 3.2.

  4 Copiadora comienza a realizar las impresiones (Ex 4.1, Ex 4.2)

  5 Se terminan las impresiones

  6 Copiadora lista para otra impresión

  Flujos Alternativos

  FA 3.1

  1. Usuario cambia el tamaño de papel.

  FA 3.2

  1. Usuario cambia el contraste

  Excepciones

  Ex 4.1. Se termina el papel y no se ha terminado de imprimir las copias

  1. Se prende indicador de "falta de papel"
  2. Adiciona el papel
  3. Continúa realizando las copias

  Ex 4.2. Atasco del papel

  1. Se prende el indicador de atasco de papel

2. Soluciona el atasco
3. Continúa realizando copias.