

2.3 Modelo Dinámico

El modelo dinámico describe el *control* del sistema, las secuencias de operaciones que ocurren en respuesta a estímulos externos.

- Se describe los objetos en término de tiempo y cambios.
- El modelo dinámico muestra patrones de actividad, organizados a través de flujos de control.
- No es importante durante esta etapa saber que hacen las operaciones, sobre que operan, o como están implementadas, solo interesa sus secuencias.
- En relación al modelo de objetos, el modelo dinámico provee una segunda dimensión a los objetos y clases de objetos, la dimensión de tiempo.
- El modelo dinámico se deriva del modelo de objetos por medio de los *estados*, representando conjuntos de valores de atributos y ligas, y *eventos*, *acciones*, y *actividades*, representando operaciones en el modelo de objetos.
- Las diferencias entre objetos se distinguen en el modelo de objeto como clases distintas, mientras que diferencias temporales en los atributos de objetos se distinguen en el modelo dinámico como estados distintos.
- Para la mayoría de los problemas, el correcto funcionamiento del sistema depende de la secuencias de las interacciones, y no de los tiempos exactos de la interacción, a menos que haya requisitos de tiempo real.
- El modelo dinámico es insignificante para un repositorio puramente de datos estáticos como una base de datos, pero sí es importante para sistemas interactivos.
- La ejecución de algoritmos no es relevante durante para el modelo dinámico si no hay manifestaciones visibles externamente.
- Un escenario corresponde a un camino particular en el diagrama de estados.
- Los diagramas de estado pueden representar como un ciclo único o un ciclo continuo:

El modelo dinámico será descrito según los siguientes aspectos:

- 1) Diagramas de estado y sus componentes
- 2) Diagramas de estado de ciclo único
- 3) Diagramas de estado de ciclo continuo
- 4) Diagrama de flujo de eventos
- 5) Descomposición de diagramas de estado
- 6) Escenario
- 7) Trazo de eventos

2.3.1 Diagramas de Estado y sus Componentes

El modelo dinámico consiste de múltiples *diagramas de estado* para todos los objetos con un comportamiento dinámico importante.

- La estructura de los patrones de *eventos*, *estados*, y *transiciones* se describe en los *diagramas de estado* de forma gráfica.
- En el *diagrama de estado* los nodos corresponden a los *estados* y los arcosdirigidos corresponden a las *transiciones* etiquetadas por los nombres de *eventos*.
- En un diagrama de estados, los *estados* representan valores particulares de los objetos.
- Los diferentes diagramas de estado en el modelo dinámico se comunican vía *eventos* comunes.
- Todas las *transiciones* entrando o saliendo de un mismo estado deben corresponder a diferentes *eventos*.
- Todos los objetos de una clase tienen el mismo comportamiento, y por lo tanto comparten el mismo diagrama de estado. Cada objeto depende de sus propias secuencias de eventos y procede de forma independiente.

Los componentes del diagrama de estado son los siguientes:

- 1) Estados
- 2) Eventos y Transiciones
- 3) Actividades
- 4) Acciones

Luego se describirá:

- 5) Caracterización de estado
- 6) Enviar eventos entre objetos

2.3.1.1 Estados

Un objeto se define como una entidad con un *estado* particular, donde el estado está definido por los valores de un subconjunto de los atributos y ligas del objeto en un momento particular.

Ejemplo: El estado del banco es solvente o insolvente, dependiendo si sus activos exceden a sus pasivos.

- Los conjuntos de valores son agrupados en un estado de acuerdo a las propiedades que afectan el comportamiento general del objeto.
- El estado especifica la respuesta de un objeto a los posibles eventos de entrada.

- La respuesta de un objeto a un evento puede variar de forma cuantitativa según los valores particulares de los atributos del objeto.
- La respuesta de un objeto a un evento puede variar de forma cualitativa para valores en diferentes estados.
- La respuesta de un objeto puede incluir un cambio de estado.
- El estado corresponde al intervalo entre dos eventos recibidos por un objeto.
- Los estados ocupan tiempo.
- Un estado puede representar implícitamente la satisfacción de algunas condiciones.
- Los estados en el modelo dinámico se derivan de los atributos definidos en el modelo de objetos.
- El estado de un objeto debe ser dividido en información:
 - externa: produce cambio de valores visibles en el objeto.
 - interna: produce cambio de valores privados, invisibles externamente.

Notación OMT

El estado se representa como una caja ovalada, donde el nombre del estado es opcional, como se muestra en la Figura 2.151.

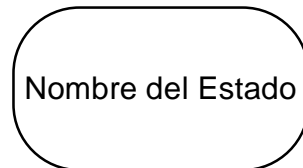


Figura 2.151 Notación de estado.

Ejemplo: En la Figura 2.152 se muestra un objeto de tipo *Computadora* el cual está *Procesando*.

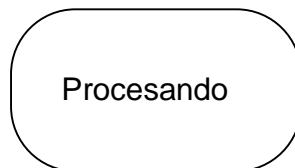


Figura 2.152 Objeto de tipo *Computadora* el cual está *Procesando*.

Existen dos notaciones especiales de estados:

2.3.1.2 Eventos y Transiciones

Un *evento* es un acontecimiento.

Ejemplo: "Un avión despegar" es un evento.

- Un evento es un estímulo individual de un objeto a otro.

Ejemplo: "Juan llama a María por teléfono".

- Un evento puede ser una *señal* de que un acontecimiento ha ocurrido.
- Los eventos pueden ocurrir de forma normal o causados por errores. La única diferencia entre ellos es la interpretación del evento.

Ejemplo: "El sistema falló"

- Las operaciones de un objeto que transmiten información son eventos, mientras que las computaciones internas no se consideran eventos, debiendo ser ignoradas.
- Un evento separa dos estados, y un estado separa dos eventos.
- La respuesta a un *evento* depende del *estado* del objeto que lo recibe, y puede incluir un cambio de *estado*, y la creación de un nuevo *evento*.
- Una *transición* se dispara cuando un *evento* ocurre. Si sale de un *estado* más de una *transición*, entonces el primer *evento* que ocurra causa su correspondiente *transición* a dispararse.
- Si ocurre un *evento* para el cual no ha sido definida una *transición* para el *estado* actual, entonces el *evento* es ignorado.
- Los eventos y estados dependen de la abstracción particular.

Ejemplo: Un agente de viajes planeando un itinerario trata cada segmento del vuelo como un solo evento; mientras que en un aeropuerto salidas y llegadas son tratadas como diferentes eventos; y para un controlador de tráfico cada vuelo consiste de varios segmentos del vuelo, cada una con su propio estado.

- Mientras los estados tienen duración y toman tiempo en completarse, los eventos representan puntos instantáneos en el tiempo.
- Los eventos en el modelo dinámico se derivan de las operaciones definidas en el modelo de objetos.

Los eventos tienen las siguientes características:

Instantaneidad: los eventos no tienen duración.

Ejemplo: "Apretar el botón izquierdo del ratón." "El vuelo 856 sale de México."

Secuencialidad: los eventos pueden preceder a otros eventos.

Un evento A debe ocurrir antes de otro evento B.

Ejemplo: "El botón izquierdo del ratón se suelta después de haber sido apretado." "El vuelo 856 debe salir de México antes de llegar a Los Angeles."

Dirección: Los eventos se transmiten en una sola dirección.

Ejemplo: Una subrutina que devuelve un resultado se considera dos eventos, el primero es la llamada a la subrutina, y el segundo es la devolución del resultado por la subrutina. Estos son dos eventos secuenciales.

Concurrencia: los eventos pueden ocurrir a la misma vez.

Dos eventos A y B ocurren sin afectarse entre ellos

Ejemplo: "El vuelo 856 que va a Los Angeles puede ir al mismo tiempo que el vuelo 842, el cual va a Dallas".

Clasificación: Se pueden clasificar los eventos en *clases de eventos*, donde cada clase de eventos indica una estructura y comportamiento común a todos sus *instancias de eventos*.

Ejemplo: "El vuelo 856 sale de México" es una instancia de "vuelo sale de", ya que todos los vuelos que salen tienen un número de vuelo particular y salen de una ciudad particular. En este caso "856" y "México" son los valores para los atributos del evento *número de vuelo* y *ciudad de salida*, respectivamente.

Los cambios, causados por eventos, en los atributos de objetos y ligas entre diferentes estados corresponden a las *transiciones de estado*.

Ejemplo: Una computadora puede pasar de un estado *Procesando* a un estado *Inerte* por medio de una transición de estados.

Notación OMT

Una transición conecta dos estados, y se representa por medio de una flecha describiendo la dirección del cambio. La notación general se muestra en el diagrama de la Figura 2.153.

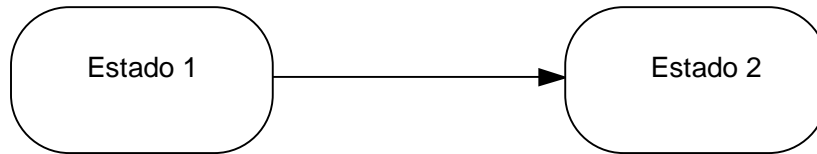


Figura 2.153 Notación para una *transición* entre estados.

Ejemplo: Un sistema de *Computadora* que pasa de un estado *Procesando* a un estado *Inactivo* se muestra en la Figura 2.154.



Figura 2.154 Transición del estado *Procesando* al estado *Inactivo*.

Los eventos se incorporan sobre las transiciones con letras cursivas, como se muestra en la Figura 2.155.

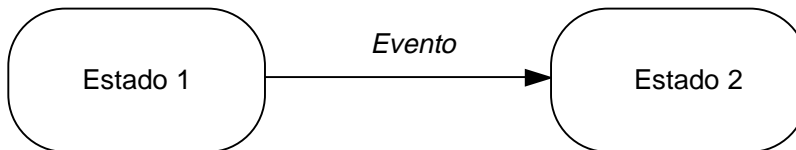


Figura 2.155 Notación para un *evento* causando una transición entre dos estados.

Ejemplo: Una *Computadora* cambia de estado de *Procesando* a *Inactivo* a raíz de la *finalización de ejecución de un proceso*, como se muestra en la Figura 2.156.

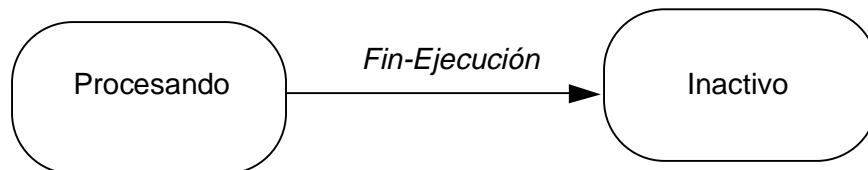


Figura 2.156 Transición del estado *Procesando* al estado *Inactivo*.

2.3.1.2.1 Atributos de Evento

- Un evento puede transferir un conjunto de valores llamados *atributos*, siendo opcional mostrar los atributos.

Ejemplo: "el vuelo sale de" contiene los atributos *aerolínea*, *número de vuelo*, *ciudad*, etc.

- Una señal no tiene atributos.
- El tiempo se considera un atributo implícito de todos los eventos.

Notación OMT

Los atributos se agregan a los eventos entre paréntesis, como se muestra en la Figura 2.157. (La lista de atributos es opcional.)

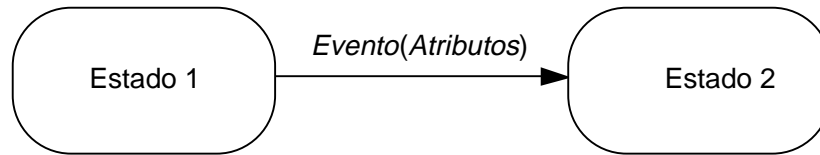


Figura 2.157 Notación para un *evento* con *atributos* causando una transición entre dos estados.

Ejemplo: Una Computadora cambia de estado de *Inerte* a *Procesando* a raíz del *comienzo de ejecución de un proceso*, el cual puede tener como atributo el nombre del proceso, como se muestra en la Figura 2.158.

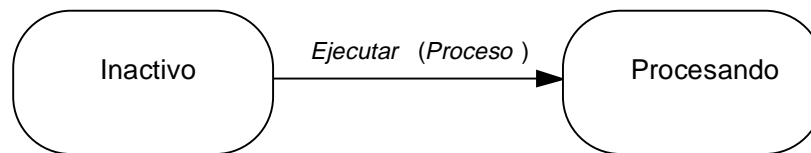


Figura 2.158 Transición del estado de *Inerte* a *Procesando* conteniendo un evento con atributos.

2.3.1.2.2 Condiciones de Transiciones

Una *condición* es una función *booleana*, con valores resultantes *verdadero* o *falso*, definida según los valores de los atributos del objeto que define el estado.

- Las condiciones son validas durante un periodo de tiempo.
- Las condiciones se aplican a las transiciones, que se dispararían solamente si el evento ocurre y la condición se cumple.

Notación OMT

Las condiciones se agregan a los eventos entre corchetes, como se muestra en la Figura 2.159. (Las condiciones, al igual que los atributos, son opcionales.)

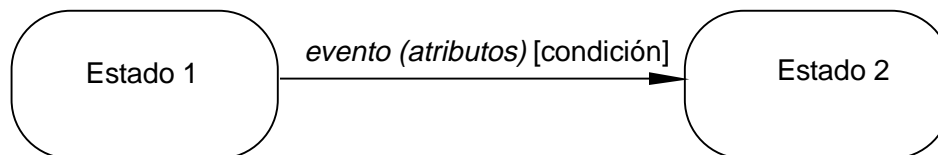


Figura 2.159 Notación para un *evento* con *condiciones* causando una transición entre dos estados.

Ejemplo: Diagrama para un semáforo, incluyendo condiciones, como se muestra en la Figura 2.160.

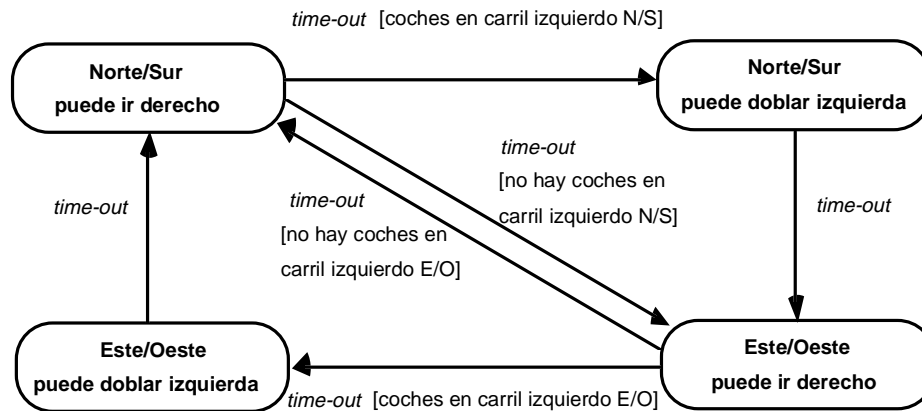


Figura 2.160. Diagrama para un semáforo, incluyendo condiciones.

2.3.1.2.3 Transiciones Automáticas

Con frecuencia el único propósito de un estado es ejecutar una actividad secuencial. Cuando la actividad se completa, una *transición automática* a otro estado se dispara.

- En los estados sin actividades las transiciones automáticas ocurren cuando el estado se alcanza.
- Se pueden asociar varias transiciones automáticas con un estado, cada uno con una condición, donde la primera condición que se satisfaga causa el disparo de la transición correspondiente.
- Si el estado tiene más de una transición automática, éstas se deben distinguir por medio de condiciones. Si ninguna de las condiciones se satisface, entonces el estado sigue activo hasta que una de las condiciones se satisfaga o hasta que un evento particular cause otra transición.

Notación OMT

Una flecha sin el nombre de un evento indica una transición automática que se dispara cuando la actividad asociada con su estado fuente se completa. Si no hay actividad, la transición no etiquetada se dispara apenas que se entra al estado. Tales transiciones también se les llama transiciones *lambda* (δ). La notación se muestra en la Figura 2.161.

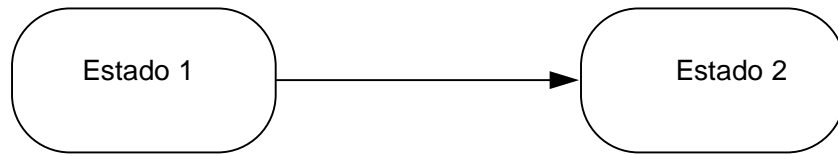


Figura 2.161 Notación para una *transición automática* entre estados, donde no existe un evento externo que cause la transición.

Ejemplo: El diagrama en la Figura 2.162 muestra una transición automática del estado *procesando-1* al estado *procesando-2*. Terminado el estado *procesando-1*, se pasa de forma automática al estado *procesando-2*.



Figura 2.162. Diagrama con transición automática entre dos procesos.

2.3.1.2.4 Transiciones Reflexivas

Es similar a una acción interna, pero causa la ejecución de acciones de entrada y salida.

Notación OMT

La notación para una transición reflexiva se muestra en la Figura 2.163.

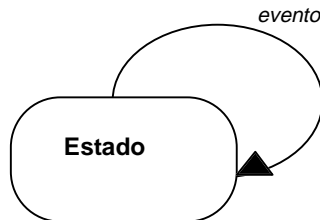


Figura 2.163 Notación para *transiciones reflexivas*.

Ejemplo: El diagrama de estado discar un número telefónico, como se muestra en la Figura 2.164, muestra transiciones reflexivas.

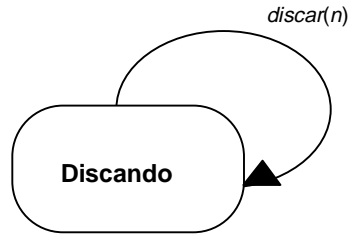


Figura 2.164 Diagrama con *transiciones reflexivas* para el discado de números telefónicos.

2.3.1.3 Actividades

Una *actividad* es una operación asociada con un estado que toma tiempo en completarse. Pueden corresponder a:

- operaciones continuas que nunca se acaban.

Ejemplo: desplegar una imagen en una pantalla.

- operaciones secuenciales que terminan luego de un intervalo.

Ejemplo: el sonido de una alarma.

- operaciones terminadas automáticamente al final de un periodo.

Ejemplo: desplegar una imagen interrumpida por una señal de borrarse.

Notación OMT

La notación para una actividad se muestra en la Figura 2.165. "do: A" ("hacer: A") dentro de una caja indica que la actividad A comienza al entrarse al estado y para al salir, o en el caso de una actividad secuencial, progresa hasta que esta se completa o es interrumpida por un evento que la termina prematuramente.

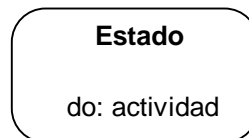


Figura 2.165. Diagrama de estados con actividades.

Ejemplo: Una Computadora que está ejecutando un proceso en el estado Procesando, como se muestra en la Figura 2.166.

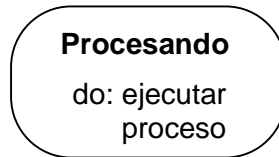


Figura 2.166. Diagrama de estados con actividades.

2.3.1.4 Acciones

Una *acción* es una operación instantánea, cuya duración es insignificante comparada a la resolución del tiempo en el sistema.

- Una acción está asociada con un evento, usándose frecuentemente para operaciones de control interno.

Notación OMT

La notación se muestra en la Figura 2.167, donde el evento, seguido de "/", preceden a la acción.

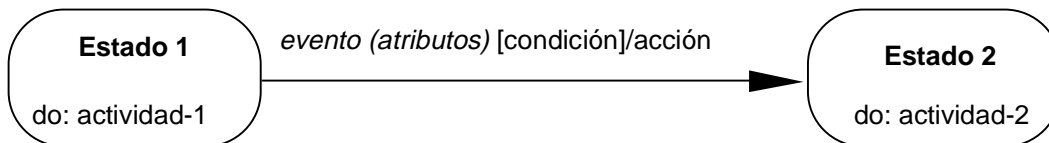


Figura 2.167. Diagrama con notación de acción.

Ejemplo: Diagrama para un menú de una ventana de computadora, conteniendo acciones, como se muestra en la Figura 2.168.

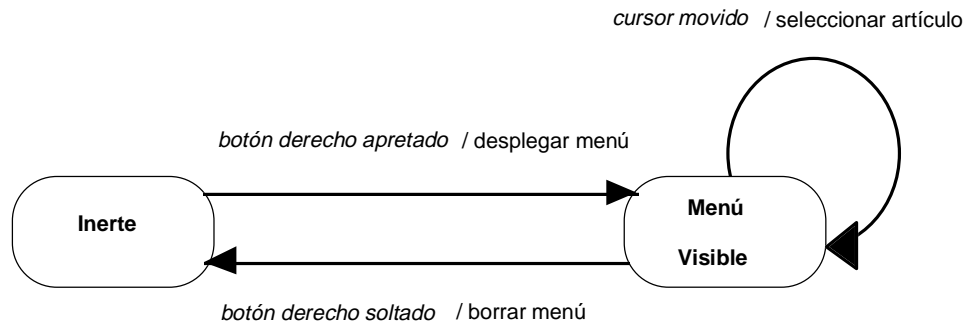


Figura 2.168. Diagrama de estados para un menú, conteniendo acciones.

2.3.1.4.1 Acciones de Entrada y Salida

Como alternativa para mostrar las acciones sobre transiciones, se puede asociar las acciones cuando se entra o se sale de un estado. La diferencia es más bien de estilo entre las dos notaciones.

- Las acciones de entrada y salida se ejecutan cuando se entra o sale de un estado, siendo útiles cuando todas las transiciones para entrar o salir de un estado comparten una acción común.
- Si se especifican operaciones múltiples en un estado, el orden de ejecución es el siguiente:
 - acciones de las transiciones que entran
 - acciones de entrada
 - actividades "do"
 - acciones de salida
 - acciones de las transiciones que salen
- Las acciones "do" pueden ser interrumpidas por eventos que causan transiciones fuera del estado, pero las acciones de entrada y salida son siempre completadas, ya que se consideran acciones instantáneas. Si se interrumpe una actividad "do", la acción de salida se ejecuta luego.

Notación OMT

Una acción de entrada es mostrada dentro de la caja de estado luego de "entry" y "/". Cuando se entra al estado con una transición, la acción de entrada se ejecuta. Es equivalente a añadir la acción a cada transición que entra. Si la transición ya tiene una acción, ésta se ejecuta primero. Las acciones de salida son menos comunes que las de entrada. Se añade luego de "exit" y "/". Cuando se sale del estado con una transición, la acción de salida se ejecuta primero. La notación para las acciones de entrada y salida se muestran en la Figura 2.169.

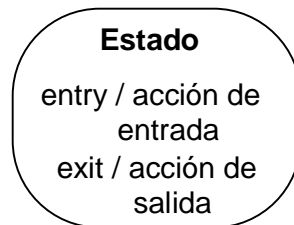


Figura 2.169 Notación para *acciones de entrada y salida*.

Ejemplo: El diagrama de estado para un control para mover la puerta del garage se muestra con acciones de transición en la Figura 2.170. El diagrama equivalente con acciones de entrada y salida se muestra en la Figura 2.171.

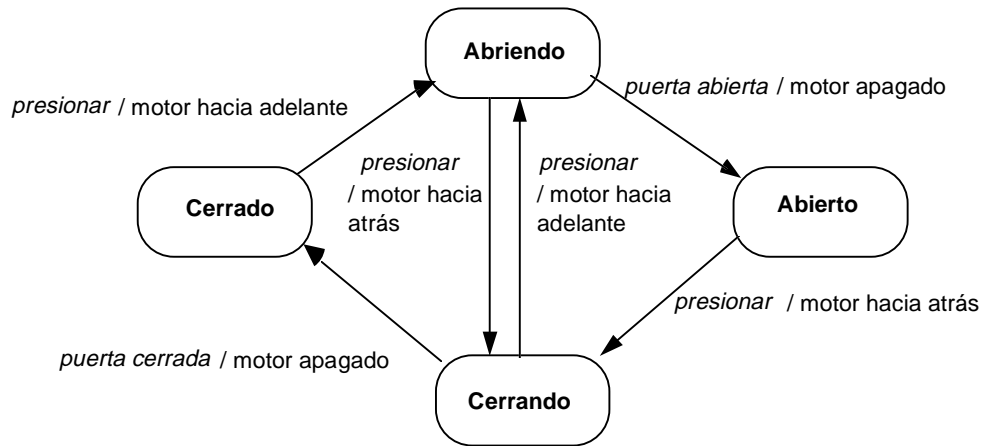


Figura 2.170. Diagrama de estados para control de la puerta del garage con acciones de transición.

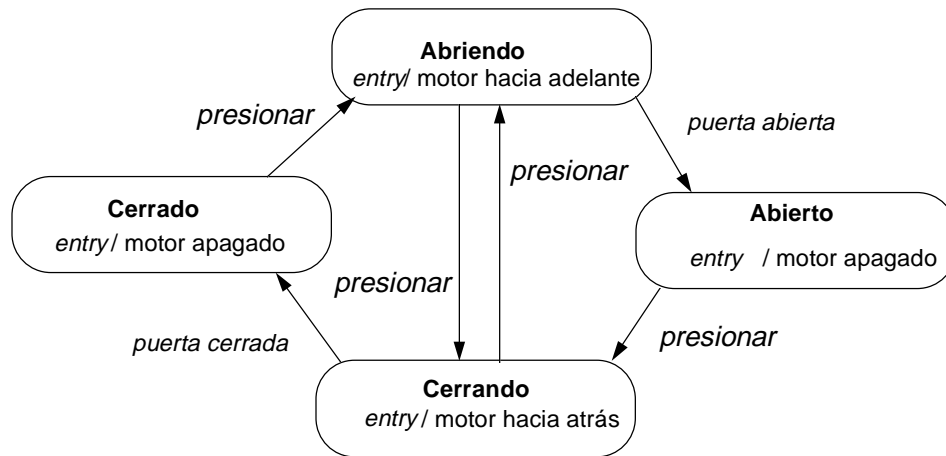


Figura 2.171. Diagrama de estados con acciones en transiciones para control de la puerta del garage con acciones de entrada y salida.

2.3.1.4.2 Acciones Internas

Un evento puede causar una acción que se ejecute sin causar un cambio de estado. Son útiles para acciones recurrentes que no resultan en cambio de estado, y se tratan como una actividad dentro del estado.

- Cuando tal evento ocurre, su acción es ejecutada pero no las acciones de entrada o salida para el estado. Hay por lo tanto una diferencia entre acciones internas y las transiciones reflexivas que causan acciones de salida y entrada para el estado que se ejecuta.

Notación OMT

El nombre del evento se escribe dentro de la caja de estado, seguido de "/" y el nombre de la acción ("entry", "exit", y "do" son palabras reservadas). La notación se muestra en la Figura 2.172.

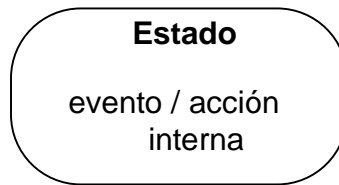


Figura 2.172 Notación para *acciones internas*.

Ejemplo: El diagrama de estado para el evento *discar(n)* conteniendo la acción *añadir número*, como se muestra en la Figura 2.173, se puede mostrar como una acción interna, como se muestra en la Figura 2.174.

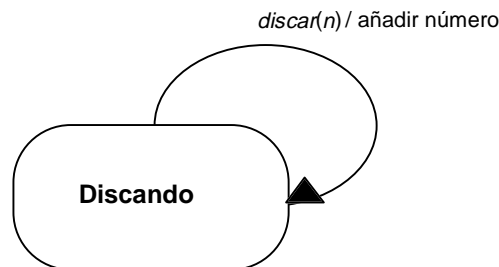


Figura 2.173 Diagrama con una transición reflexiva, conteniendo el evento *discar(n)* y la acción *añadir número*, para el discado de números telefónicos.

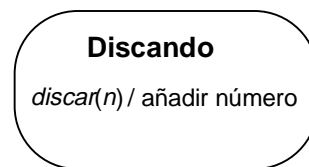


Figura 2.174 Diagrama con una acción interna, conteniendo el evento *discar(n)* y la acción *añadir número*, para el discado de números telefónicos.

2.3.1.5 Caracterización de Estado

La *caracterización de estado* describe el estado de un objeto según los valores de atributos y ligas, sus condiciones, y los eventos que acepta, incluyendo acciones a tomar y próximos estados.

- Se puede caracterizar el estado con una descripción en lenguaje natural.
- Se incluye en la caracterización del estado:
 - nombre: nombre del estado que debe ser lo más descriptivo posible.
 - descripción: el propósito del estado resumido en lenguaje natural.
 - eventos predecesores: la secuencia de eventos para arribar al estado.
 - condiciones: especifican funciones booleanas de los atributos que caracterizan el estado.
 - tabla estímulo-respuesta: se describe estados actuales y próximos. Se lista en una tabla los posibles eventos sucesores, acciones resultantes, y estados sucesores.

Ejemplo: Caracterización de estado de una alarma sonando, como se muestra en la Figura 2.175.

Estado: <i>Alarma Sonando</i>		
Descripción: alarma del reloj suena para indicar la hora deseada		
Secuencia de eventos predecesores al estado:		
<ul style="list-style-type: none"> <i>poner alarma</i> (hora deseada) cualquier secuencia sin incluir <i>borrar alarma</i> hora actual = hora deseada 		
Condición que caracteriza el estado:		
<ul style="list-style-type: none"> alarma = on, y hora deseada \leq hora actual \leq hora deseada + 20 segundos, y ningún botón ha sido apretado desde la hora deseada 		
Eventos aceptados en el estado:		
evento	acción	próximo estado
hora actual = hora deseada + 20	borrar alarma	normal
botón apretado	borrar alarma	normal

Figura 2.175. Caracterización de estado para una alarma sonando.

2.3.1.6 Enviar Eventos entre Objetos

Un objeto puede ejecutar la acción de enviar un evento a otro objeto.

- La acción "enviar $E(\text{atributo})$ " envía el evento E con *atributos* a otro objeto u objetos. (La palabra enviar puede ser omitida si es claro que E es el nombre del evento.)
- Si un estado acepta eventos de más de un objeto, el orden en el cual los eventos concurrentes son recibidos puede afectar el estado final. Se llama a esto, una *condición de carrera* (*race condition*).
- Cuando un objeto interactúa con otro objeto, enviar un evento es casi lo mismo que ejecutar una acción.

Notación OMT

La notación general para las acciones de entrada y salida se muestra en la Figura 2.176.

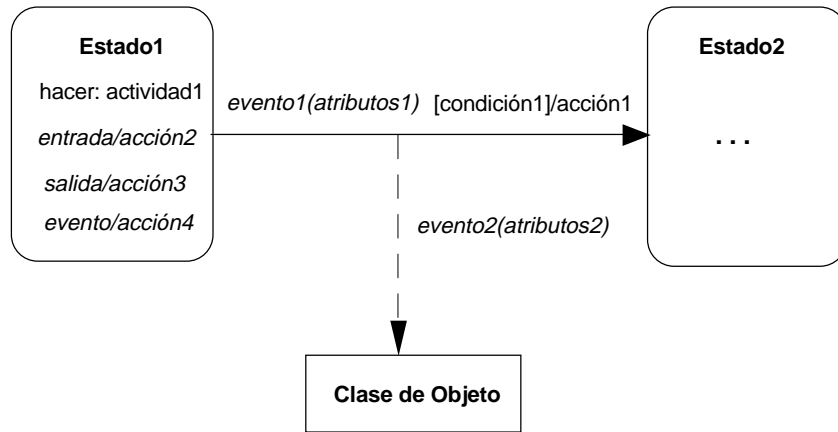


Figura 2.176. Diagrama de estados con notación completa, incluyendo envío de eventos entre objetos.

2.3.2 Diagramas de Estado de Ciclo Unico

- Los *diagramas de un sólo ciclo* representan objetos con vidas finitas, con *estado inicial* y *estado final*. El estado inicial se da cuando se crea el objeto y el estado final cuando se destruye.

Notación OMT

El *estado inicial* describe el punto de comienzo de un diagrama que describe los estados de un objeto. La notación se muestra en la Figura 2.177.

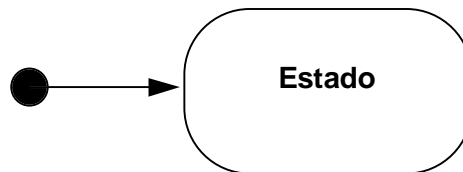


Figura 2.177 Estado inicial de un diagrama.

El *estado final* describe el punto final de un diagrama que describe los estados de un objeto. La notación se muestra en la Figura 2.178.

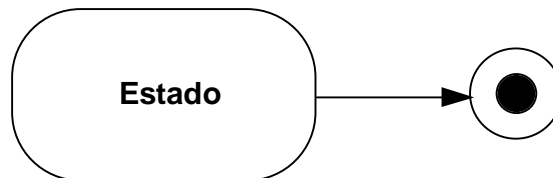


Figura 2.178 Estado final de un diagrama.

Ejemplo: Una *Partida de Ajedrez*, como se muestra en la Figura 2.179, consiste de un estado inicial donde las blancas juegan, y varios estados finales dependiendo de si ganan las blancas, las negras, o empatan.

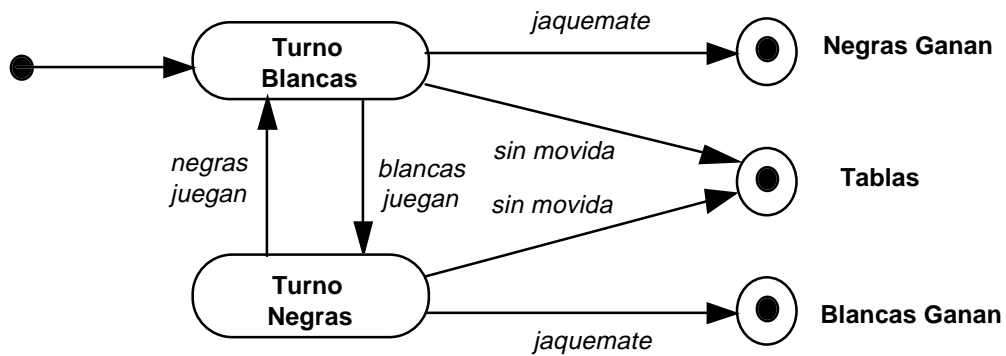


Figura 2.179. Diagrama de estados de un sólo ciclo para una *partida de ajedrez*.

2.3.3 Diagramas de Estado de Ciclo Continuo

- Los *diagramas de ciclo continuo* representan secuencias de operaciones que regresan a algún punto inicial de donde se vuelven a continuar las mismas secuencias de eventos y estados. Pueden no incluir estados iniciales (ni finales).

Ejemplo: El diagrama de estados para la *llamada telefónica* donde siempre se regresa al estado *inactivo*, como se muestra en la Figura 2.180.

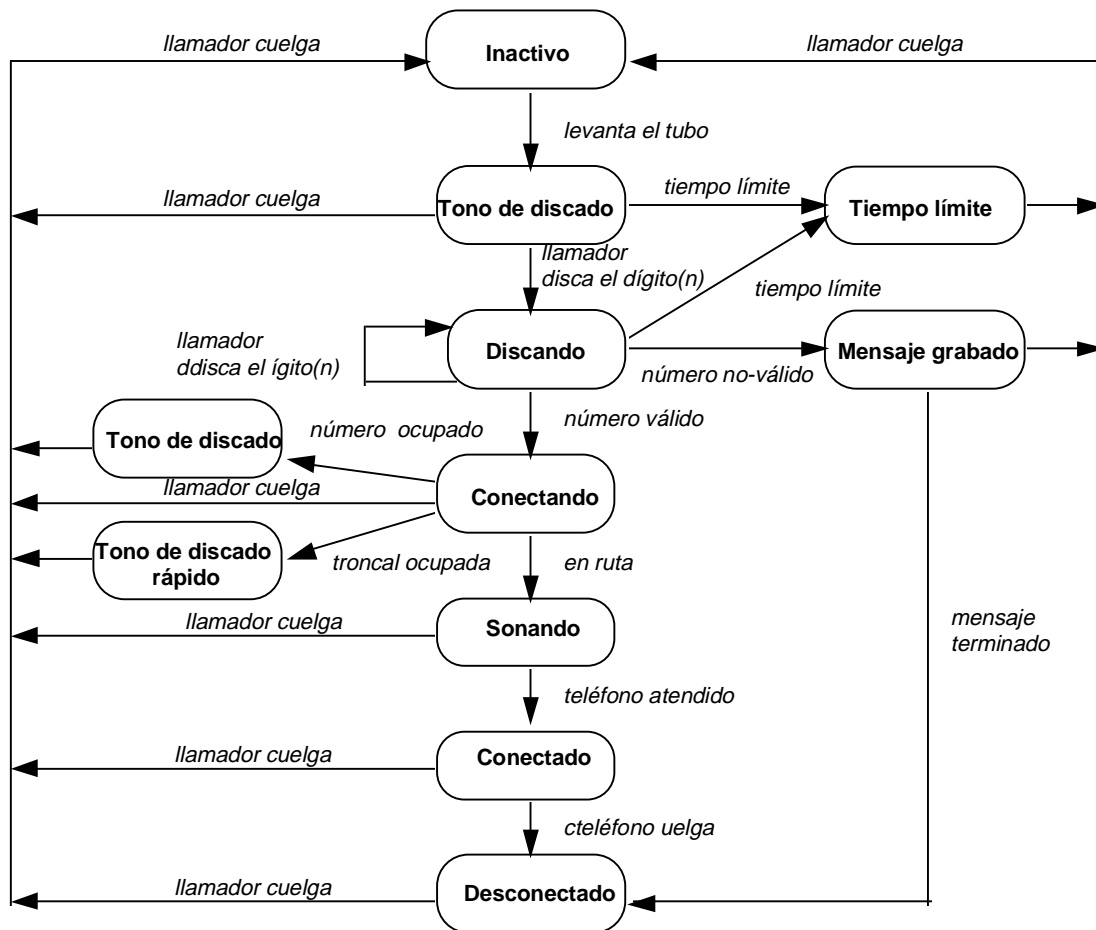


Figura 2.180. Diagrama de estados de ciclo continuo para una llamada telefónica

2.3.4 Diagrama de Flujo de Eventos

El diagrama de flujo de eventos hace un resumen de los eventos entre clases sin importar su secuencia.

- Se incluye los eventos de todos los escenarios, incluyendo los eventos de error.
- Se utiliza para mostrar el flujo de eventos entre los diferentes *módulos* del sistema.

Notación OMT

Se representan los objetos como cajas, donde los eventos son líneas de interconexión. Se listan todos los eventos del mismo productor al mismo consumidor en la misma flecha.

Ejemplo: El diagrama de flujo de eventos para una llamada telefónica se muestra en la Figura 2.181.

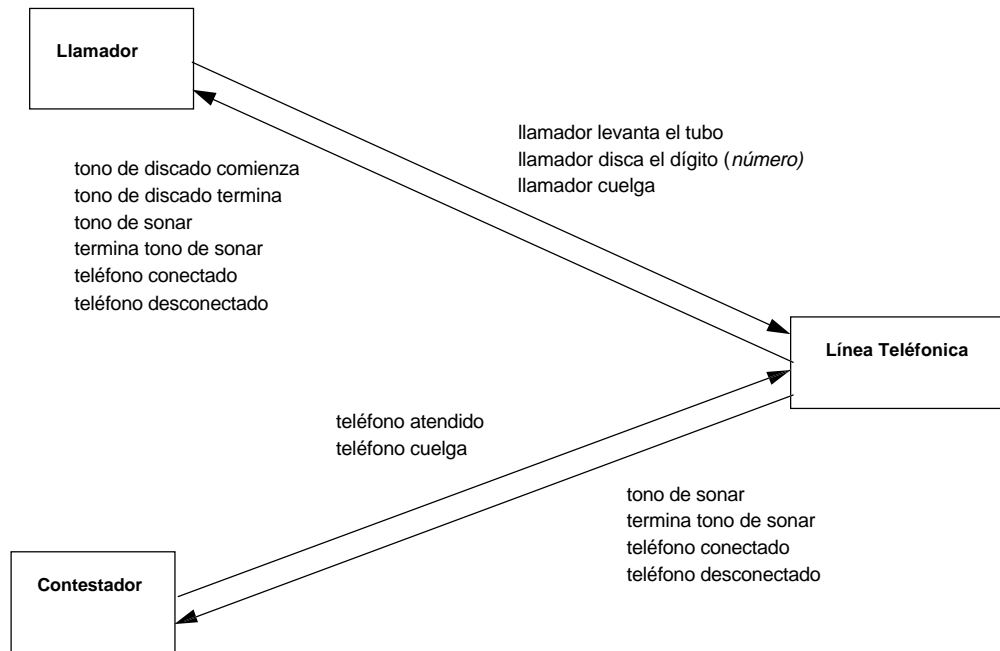


Figura 2.181. Diagrama de flujo de eventos para una llamada telefónica al número "555-1234".

2.3.5 Descomposición de Diagramas de Estados

Las actividades anidadas se utilizan para describir en mayor detalle los diagramas de estado de más alto nivel.

- Los subdiagramas se describen como diagramas de un sólo ciclo, con transiciones de entrada y salida.

Pueden expandirse:

- 1) actividades
- 2) transiciones

Notación OMT

Ejemplo: El diagrama de estados de alto nivel para una máquina vendedora se mostró en la Figura 2.182.

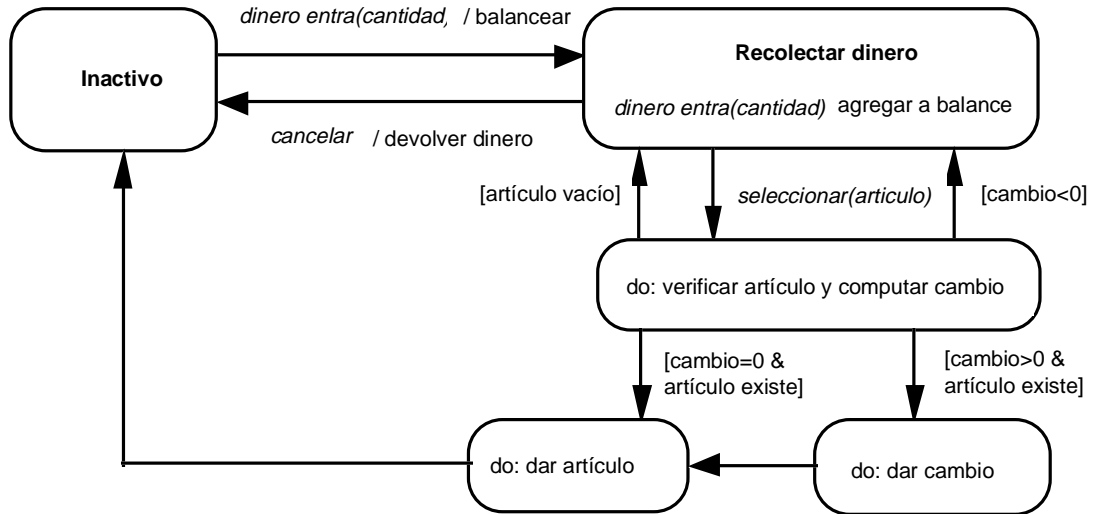


Figura 2.182. Diagrama para una máquina vendedora.

Expansión de actividad: En la Figura 2.183 se muestra el diagrama de más bajo nivel para la actividad *dar artículo*.

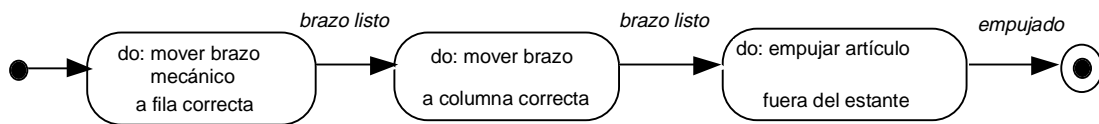


Figura 2.183. Diagrama de estados para la actividad *dar artículo* para una máquina vendedora.

Expansión de transición: En la Figura 2.184 se muestra el diagrama de más bajo nivel para la transición *seleccionar(artículo)*.

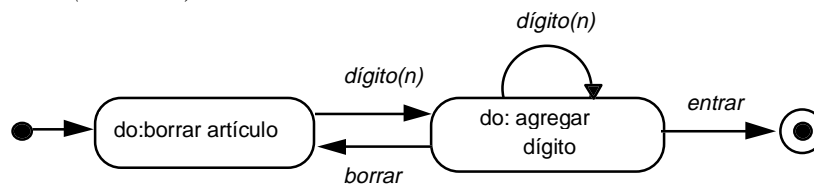


Figura 2.184. Diagrama de estados para la transición *seleccionar artículo* para una máquina vendedora.

Los diagramas de estado también se pueden estructurar aprovechando los conceptos de:

- 1) generalización
- 2) agregación
- 3) concurrencia

2.3.5.1 Generalización

La generalización es equivalente a expandir actividades.

- Permite describir una actividad a un nivel alto y expandirla a un nivel más bajo añadiendo detalles.
- Permite que los estados y eventos se organicen en jerarquías.

2.3.5.1.1 Generalización de Estado

Un objeto en un diagrama de estado de nivel alto debe estar exactamente en un estado en el diagrama anidado.

- La generalización de estado es una condición lógica "o", donde el diagrama de alto nivel debe estar en uno de los estados en los diagramas de más bajo nivel.
- Los estados en el diagrama anidado son refinaciones del estado del diagrama de alto nivel.
- Se usan diagramas de alto nivel para nombrar los superestados.
- En los diagramas de más bajo nivel se capturan los subestados.
- Cualquier transición o acción que se aplica a un estado, es heredado por todos sus subestados, a menos que se sobre-escriba con una transición equivalente en el subestado.

Notación OMT

Ejemplo: El diagrama de estados anidados para la *transmisión automática* de un *automóvil* se muestra en la Figura 2.185.

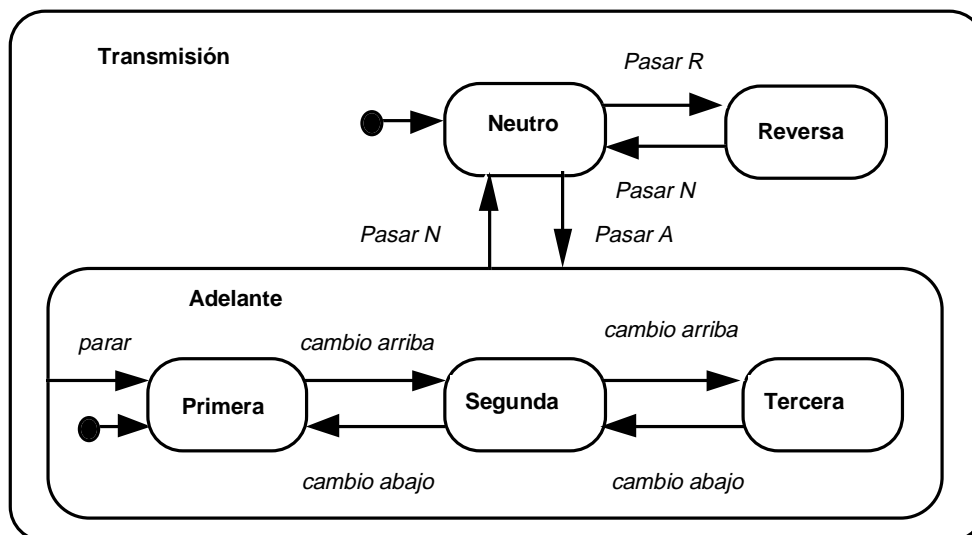


Figura 2.185. Diagrama de estados anidados para la *transmisión automática* de un *automóvil*.

- Los bordes de un subdiagrama dentro del diagrama de más alto nivel se conoce como *contornos*. Las transiciones del contorno a un subestado representan transiciones heredadas

por los subestados. En el caso de transiciones explícitas entre subestados y estados en el contorno o fuera, todos los estados deben aparecer en el mismo diagrama usando la notación de contorno. En casos más simples, cuando no hay interacción excepto para la inicialización y terminación, los diagramas anidados pueden ser dibujados en un diagrama separado y referido con el nombre de la actividad.

2.3.5.1.2 Generalización de Evento

Los eventos se pueden organizar en una jerarquía de generalizaciones con herencia de los atributos de evento.

Notación OMT

Ejemplo: La Figura 2.186 muestra parte de un árbol de eventos de entrada para una estación de trabajo. Un evento de entrada genera transiciones en cualquier ancestro del tipo de evento. Por ejemplo "a" generaría una transición en el evento *alfanumérico* al igual que en *caracter de teclado*.

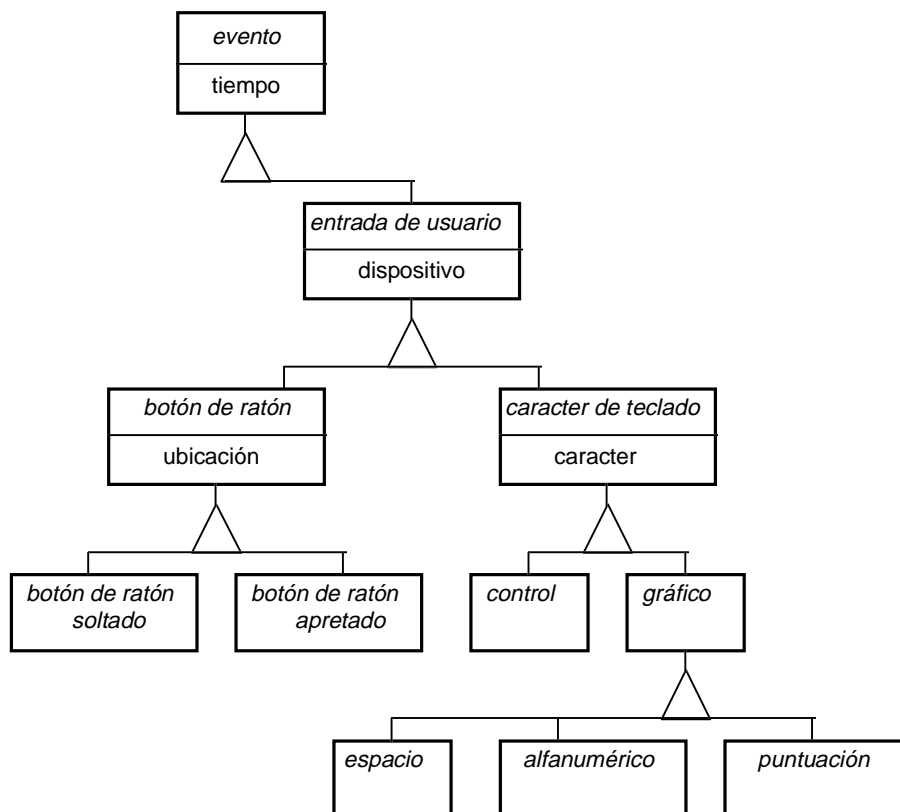


Figura 2.186. Jerarquía de eventos parcial para eventos de teclado.

2.3.5.2 Agregación

La *agregación* permite que un estado se divida en componentes ortogonales con interacción limitada, siendo equivalente a los estados concurrentes, permitiendo la interacción de partes dentro de un agregado de objetos.

Notación OMT

Ejemplo: El diagrama de clases para un agregado describiendo el funcionamiento de un automóvil es mostrado en las Figuras 2.187.

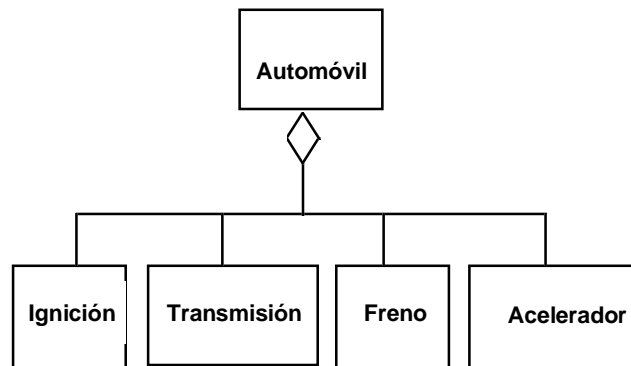


Figura 2.187. Diagrama de clases para un *automóvil*.

El diagrama de estados para el resto de los componentes del agregado son mostrados en las Figuras 2.188 y 2.189.

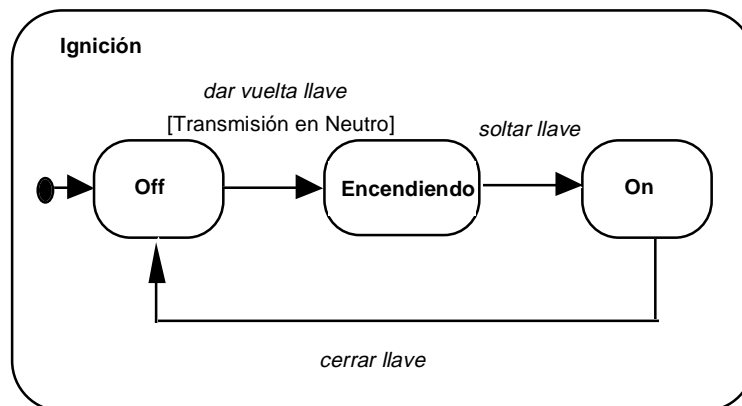


Figura 2.188. Diagrama de estados para la *ignición* de un *automóvil*.

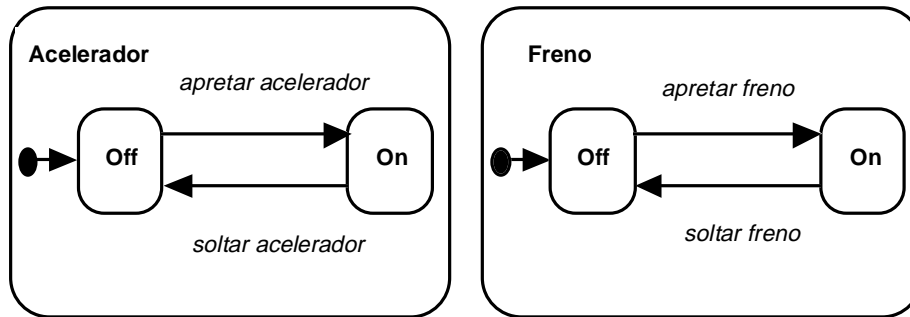


Figura 2.189. Diagrama de estados para el *acelerador* y *freno* de un *automóvil*.

2.3.5.3 Concurrencia

Se analizarán los siguientes puntos con respecto a la concurrencia:

- 1) Concurrencia por agregación
- 2) Concurrencia por atributos dentro de un objeto
- 3) Concurrencia por actividades dentro de un objeto

2.3.5.3.1 Concurrencia por Agregación

El estado de un sistema completo es representado por el producto de todos los estados de todos los objetos, donde el número de objetos puede variar de forma dinámica.

- El diagrama de estado para un ensamblado es una colección de diagramas de estado, uno para cada componente.
- El estado compuesto es la combinación de estados de todos sus subdiagramas, cada componente dispara sus transiciones en paralelo, según sus condiciones.
- Cada diagrama de estado se puede tratar de forma independiente, pero puede interactuar con otros diagramas de estado.

Notación OMT

Ejemplo: El agregado de automóvil, mostrado en las figuras anteriores, es un ejemplo de la concurrencia en estados en un agregado.

2.3.5.3.2 Concurrencia por Atributos dentro de un Objeto

La concurrencia dentro del estado de un objeto se puede dar cuando los atributos de los objetos se pueden partir en subconjunto de estados, cada uno con su propio diagrama de estado.

- El estado del objeto está formado por un estado de cada subdiagrama, que no tiene que ser independiente.
- El mismo evento puede causar transiciones en más de un subdiagrama.

Notación OMT

La concurrencia dentro de un sólo estado compuesto de un objeto es descrito por subdiagramas con líneas punteadas, mientras que el estado compuesto se puede escribir en una región separada por líneas solidas.

Ejemplo: El diagrama de estado con concurrencia para un *partido de tenis* se muestra en la Figura 2.190. Cuando un jugador gana un "set", ese jugador se encuentra *arriba* mientras que el otro se mantiene *abajo*. El jugador que gana primero 2 "sets" gana el partido. El estado inicial de ambos subdiagramas es *abajo*. Los estados *1 gana* y *2 gana* no son estados concurrentes ya que solo un jugador puede ganar, por lo cual están fuera del diagrama *Partido de Tenis*.

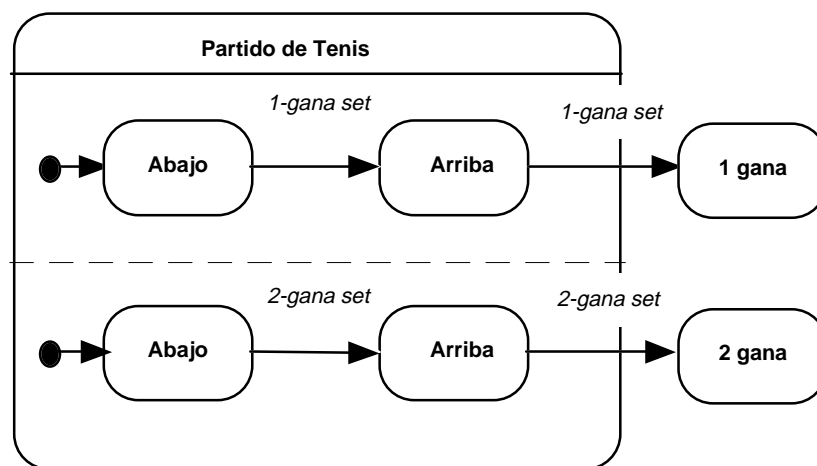


Figura 2.190. Diagrama de estados con concurrencia para un *partido de tenis*.

2.3.5.3.3 Concurrencia por Actividades dentro de un Objeto

La concurrencia dentro del estado de un objeto se puede dar cuando dos actividades se pueden activar de forma concurrente, partiendo el estado del objeto en subconjunto de estados, cada uno con su propio diagrama de estado.

- Aunque los pasos internos no están necesariamente sincronizados, las dos actividades deben ser completadas antes que el objeto entre en su siguiente estado.
- La actividad compuesta asume exactamente un estado en cada subdiagrama.
- Cualquier transición a un estado con subdiagrama concurrente activa cada uno de los subdiagramas. Si alguno de los subdiagramas es omitido de la transición, se comienza en el estado por omisión inicial del subdiagrama.
- El estado siguiente queda activado cuando ambos eventos ocurren, que no tienen que ser simultáneamente.

Notación OMT

La concurrencia dentro de un sólo estado compuesto de un objeto es descrito por subdiagramas con líneas punteadas, mientras que el estado compuesto se puede escribir en una región separada por líneas solidas.

Ejemplo: Máquina de Cajero Automático, como se muestra en la Figura 2.191. El evento *listo* causa una transición del estado *Preparando* a los dos subestados de *Emitiendo*, los cuales se sincronizan mediante los dos eventos *dinero tomado* y *tarjeta tomada* antes de entrar en el siguiente estado *Listo para borrar*.

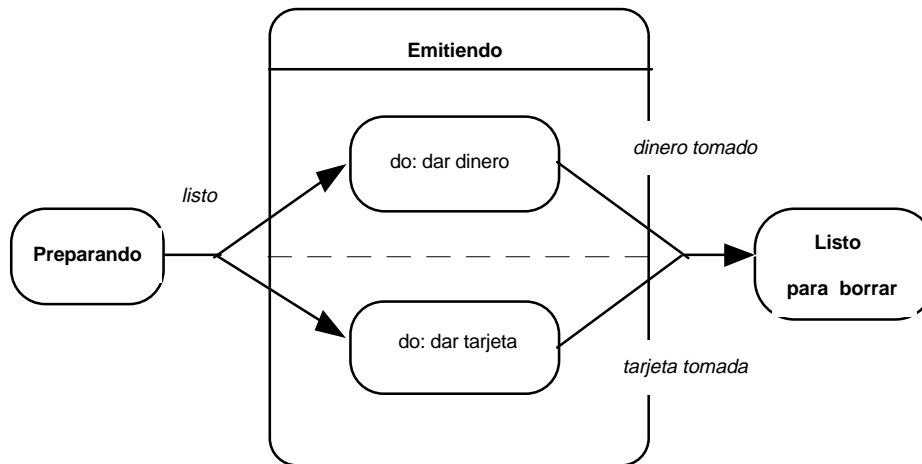


Figura 2.191. Diagrama de estado con sincronización para una máquina de cajero automático.

2.3.6 Escenarios

Un *escenario* muestra un diálogo típico del sistema, que ocurre cuando la información es intercambiada entre un objeto del sistema y un agente externo (como usuario, dispositivos externos, etc.), pudiendo incluir parte o todos los eventos del sistema.

- Los escenarios son creados según la operación particular del sistema, y son utilizados para verificar, validar, y examinar las condiciones de error del sistema.
- Cada vez que la información entra o sale del sistema un evento ocurre.
- Eventos sin parámetros son importantes y muy comunes. La información en tal evento es el hecho de que el evento ocurrió.
- Para cada evento, se debe identificar el objeto (que puede ser un agente externo) que causa el evento, y los parámetros del evento.

Ejemplo: El evento *número discado* tiene el valor del *número* como parámetro.

Notación OMT

Ejemplo: El escenario para una llamada telefónica "normal" (al número 555-1234) se muestra en la Figura 2.192. (Los valores para los atributos de los eventos se muestran entre paréntesis.)

llamador levanta el tubo
tono de discado comienza
llamador disca el dígito (5)
tono de discado termina
llamador disca el dígito (5)
llamador disca el dígito (5)
llamador disca el dígito (1)
llamador disca el dígito (2)
llamador disca el dígito (3)
llamador disca el dígito (4)
teléfono marcado comienza a sonar
tono de sonar aparece en el teléfono que llama
teléfono marcado es atendido
teléfono marcado deja de sonar
tono de sonar desaparece en el teléfono que llama
los teléfonos están conectados
teléfono marcado cuelga
los teléfonos están desconectados
teléfono que llama cuelga

Figura 2.192. *Escenario* para una llamada telefónica al número "555-1234".

2.3.7 Trazo de Eventos

Un *trazo de eventos* es un escenario aumentado, describiendo las secuencias de eventos y clarificando las interacciones entre los objetos. Se determinan los consumidores y productores de los eventos.

Notación OMT

Se muestra cada objeto como una línea vertical y cada evento como una flecha horizontal entre los diferentes objetos. El momento durante el cual ocurre un evento se incrementa de arriba hacia abajo, donde solo importa las secuencias temporales de los eventos y no el momento exacto durante el cual ocurren. Los eventos pueden ocurrir de forma simultánea.

Ejemplo: El trazo de eventos para una llamada telefónica se muestra en la Figura 2.193.

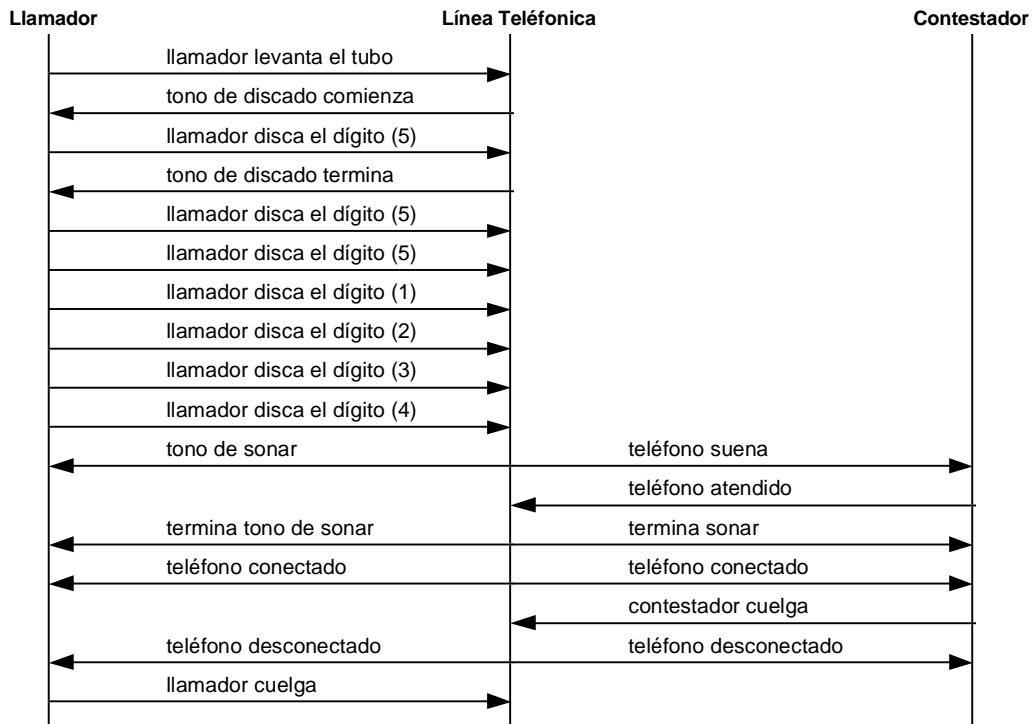


Figura 2.193. Trazo de eventos para una llamada telefónica al número "555-1234".