

Seminario UML

Diagramas de actividad y diagramas de estados

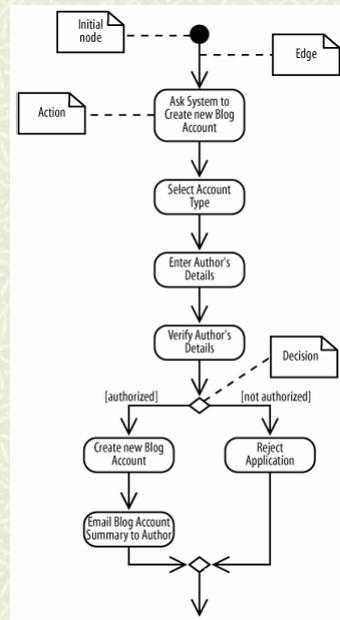


J.M. Drake

Notas:

Elementos básicos de un diagrama de actividad

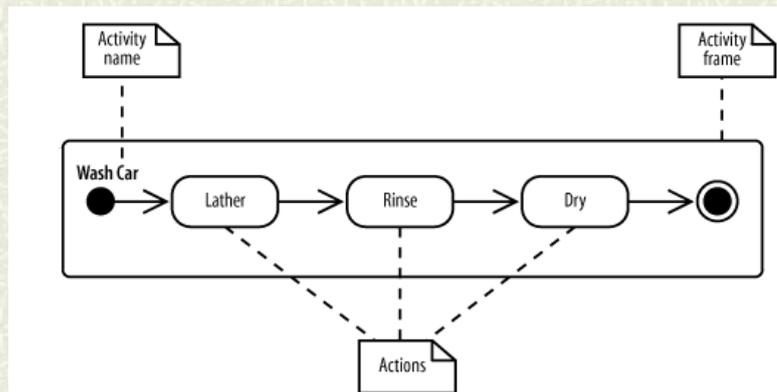
- ✦ Los diagramas de actividad permiten describir como un sistema implementa su funcionalidad.
- ✦ Los diagramas de actividad modelan el comportamiento dinámico de un procedimiento, transacción o caso de uso haciendo énfasis en el proceso que se lleva a cabo.
- ✦ Los diagramas de actividad es uno de los elementos de modelado que son mejor comprendidos por todos, ya que son herederos directos de los diagramas de flujo.
- ✦ Los diagramas de actividad son mas expresivo que los diagramas de flujo. También heredan características de:
 - Los diagramas de estado.
 - Los diagramas de flujo de datos.
 - Las redes de Petri.



Notas:

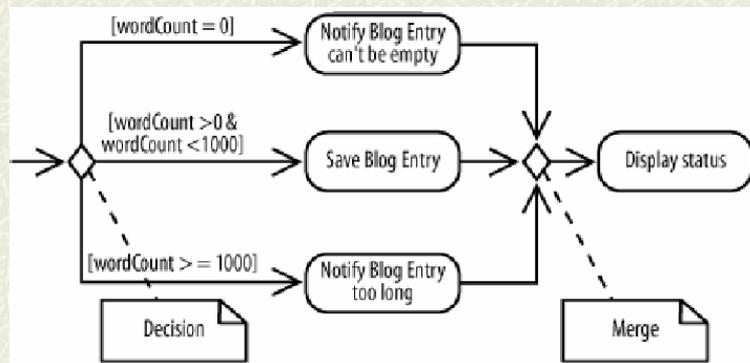
Actividades y acciones

- ✦ Una acción es un paso de un proceso que tiene la semántica “run to completion” (Se inicia para ser terminado)
- ✦ Una actividad es un conjunto de acciones que modelan un proceso. No tiene la semántica “run to completion”. Una actividad se modela mediante un diagrama de actividad.
- ✦ Enjabonar, enjuagar o secar un coche son acciones de la actividad “Lavar un coche”



Notas:

Branching and merges

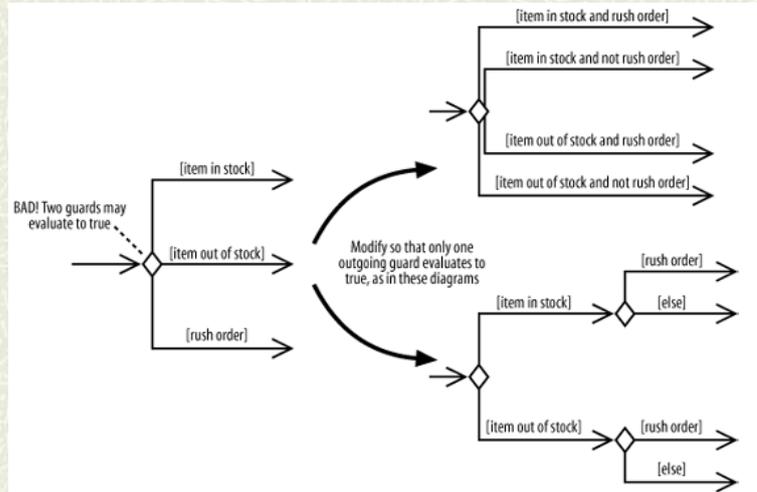


- # Las decisiones representan las alternativas de flujo de control en un diagrama que se llevan a cabo en función de una condición.
- # Las condiciones de guarda asociadas a cada rama de salida determinan la opción de flujo de control que se sigue.
- # Las ramas de flujo de control abiertas en una o varias condiciones se cierran en un punto de convergencia (merge).

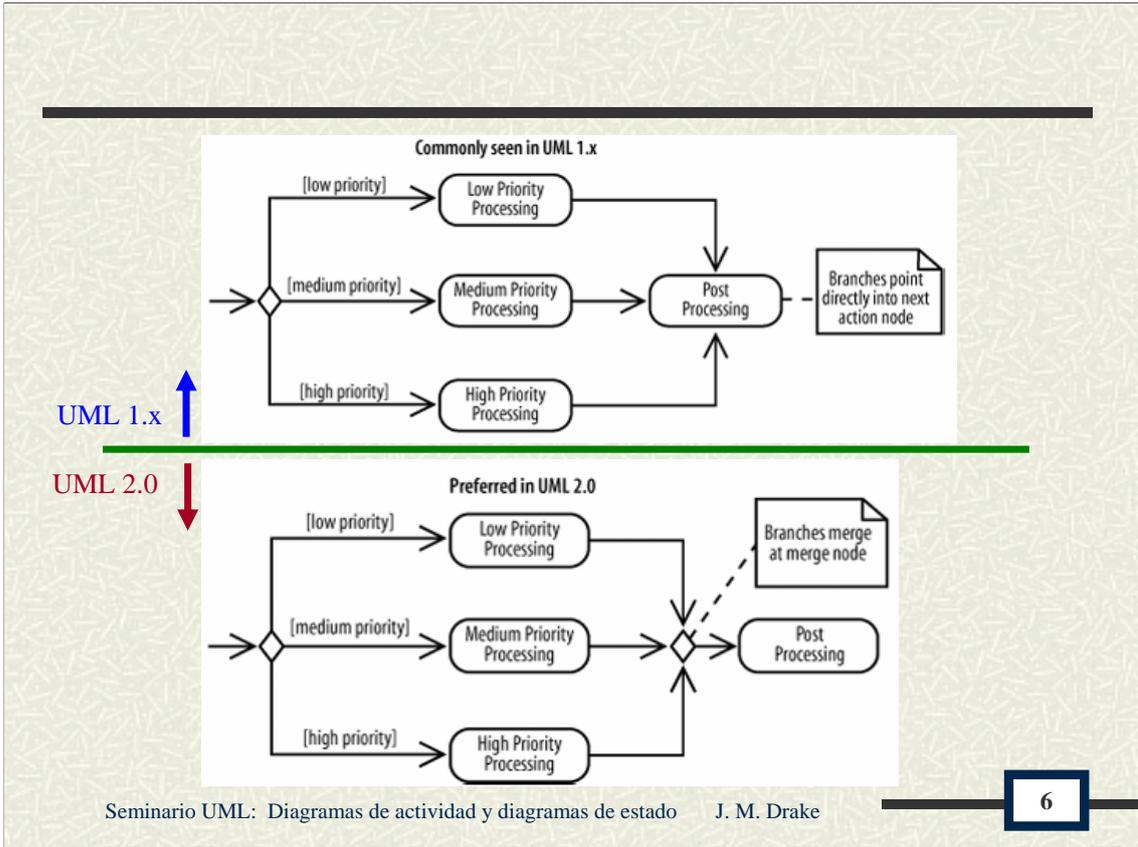
Notas:

Decisiones consistentes

- La opciones de una decisión deben ser:
 - Completas.
 - No ambiguas.

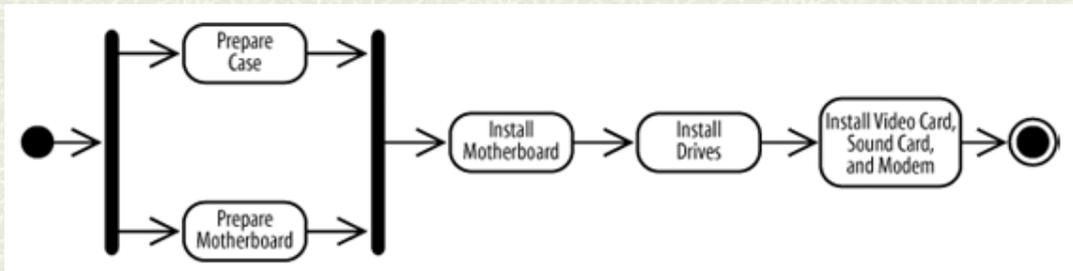


Notas:



Notas:

Fork y Joint

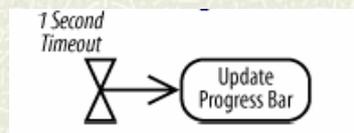
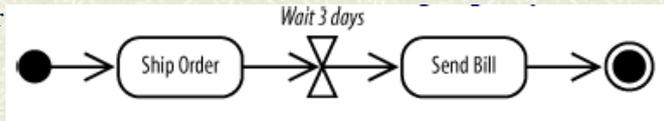


- # Los fork y los joint se utilizan en los diagramas de actividad para describir concurrencia entre acciones o actividades.
- # Las líneas de flujo de salida de un fork representa líneas de ejecución que se ejecutan concurrentemente.
- # Las líneas de flujo de entrada de un joint se sincronizan para continuar en una única línea de flujo. Todas la acciones de las líneas de flujo previas a un joint deben completarse antes de que se ejecute la primera acción de la línea posterior a él.

Notas:

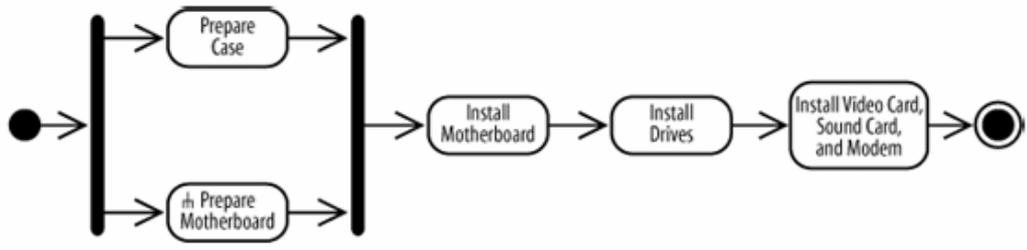
Time events

- Los eventos de tiempo modelan:
 - Activaciones temporizadas.
 - Timeouts
 - Retrasos.
 -
- Un evento de tiempo puede ser el inicio de una actividad.
- Un evento temporizado con flujo de entrada representa una única activación tras llegarle el flujo
- Un evento de tiempo sin flujo de entrada representa una activación que puede ser repetida en el tiempo.

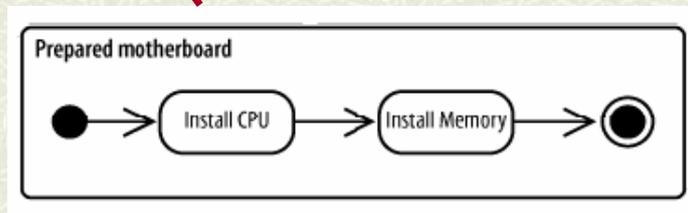


Notas:

Calling other activities

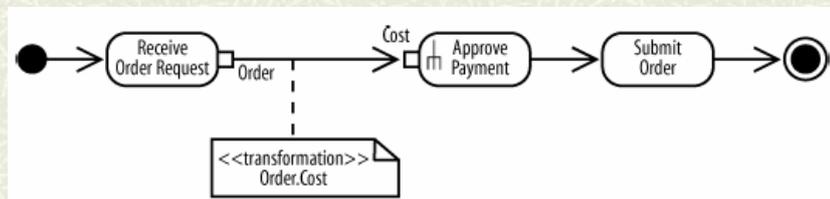
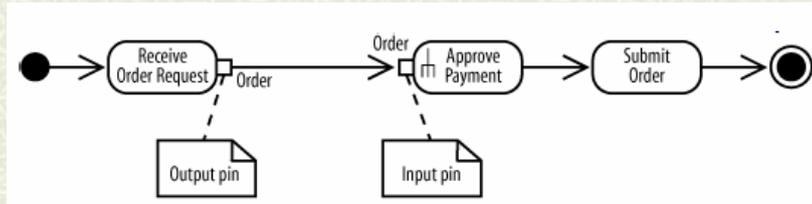
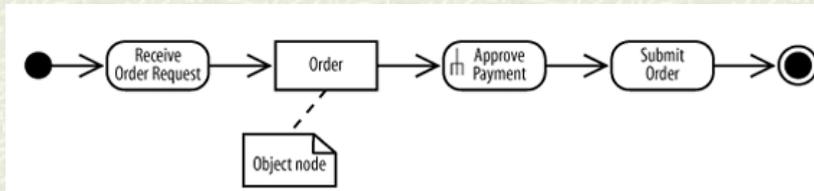


Una actividad representada en un diagrama representa un punto en el que se ejecuta la actividad descrita en otro diagrama



Notas:

Objects



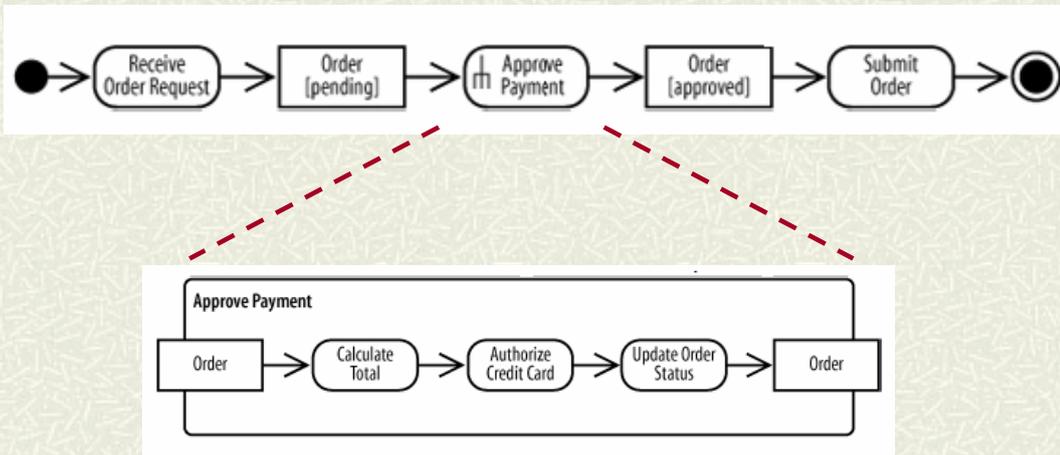
Notas:

En un diagrama de actividad se pueden representar los objetos de datos que se generan, se consumen o se intercambian en un proceso y que son relevantes para su descripción.

Cuando un objeto de datos se representan como una caja, significa que esos datos existen en el punto de flujo de control en que se insertan.

Cuando un objeto de datos se representa mediante unos pines asociados a las acciones o actividades, representan objetos de datos de entrada o de salida.

Object change state in an activity



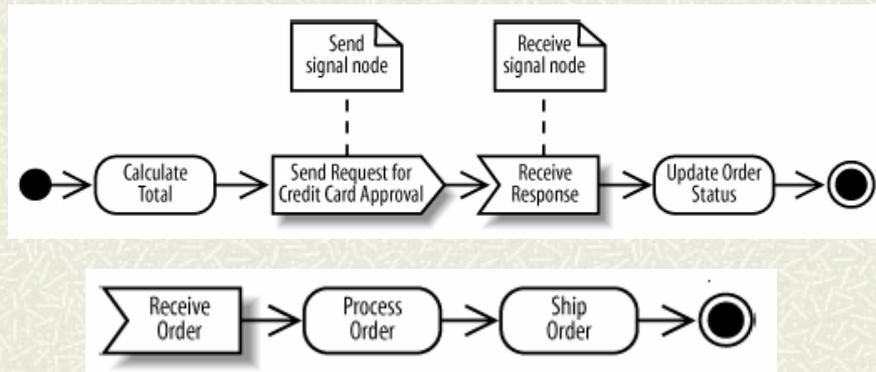
Notas:

En un diagrama de actividad, los objetos de datos pueden representar los puntos de inicio y finalización de la actividad que representa.

Sending and receiving signals

En un diagrama de actividad las señales representan interacciones del proceso que se Describe con operadores sistemas u otros procesos externos a él.

- Cuando un receptor de señales tiene flujo de entrada representa que cuando el flujo le llega se habilita para aceptar una única señal.
- Cuando no tiene flujo de entrada, representa que puede aceptar uno o muchas señales.



Notas:

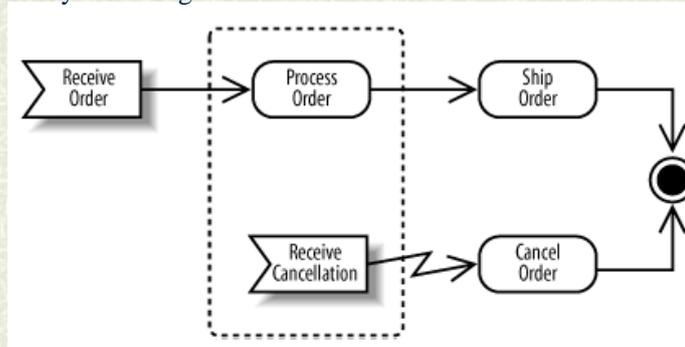
Starting an activity

- ✦ Una actividad se puede iniciar por:
 - Cuando se invoca de forma regular: se representa mediante un círculo.
 - Cuando se recibe un objeto de dato de entrada
 - Cuando se produce un evento temporizado
 - Cuando se recibe una señal externa.

Notas:

Interrupting an activity

Una actividad de duración no atómica puede concluirse por la ocurrencia de un evento o una señal externa. Para ello se define una región de interrupción mediante una línea que engloba las actividades o acciones que pueden ser interrumpidas por el evento o señal que también se incluye en la región.

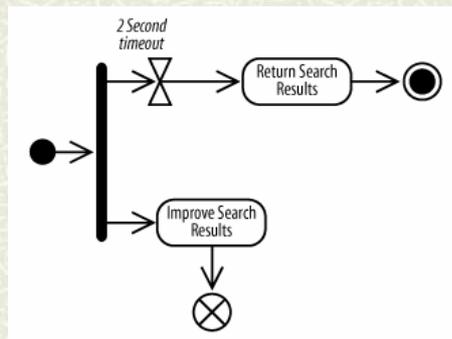


Notas:

Ending a flow

UML 2.0 permite concluir una línea de control de un flujo concurrente sin terminar la Actividad:

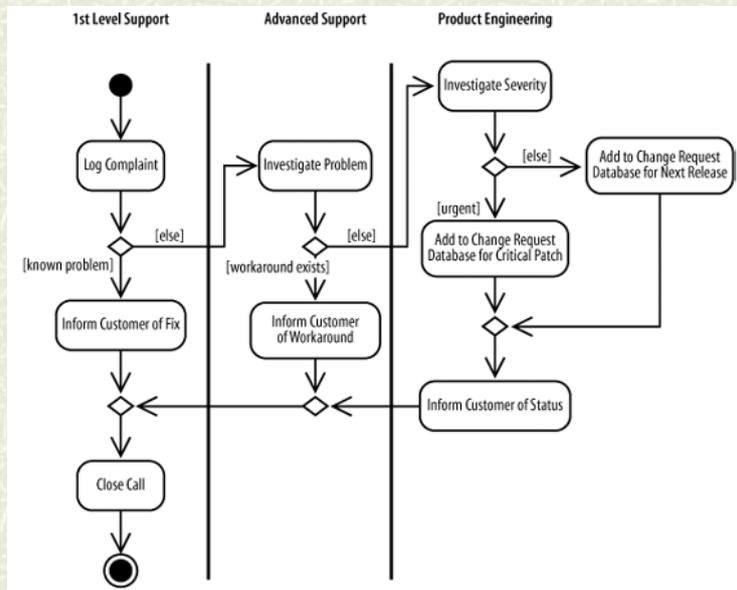
- Un ojo de buey termina la actividad completa (todos los threads).
- Un círculo con un aspa, representa la finalización de una línea de flujo (thread).



Notas:

Partitions or swimlanes

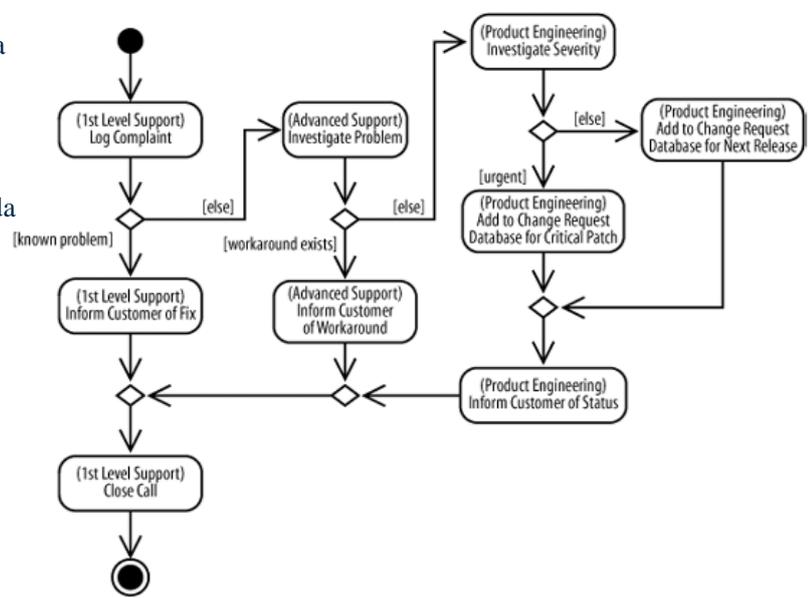
Los swimlanes representan los procesos, participantes o elementos responsables de ejecutar un conjunto de acciones.



Notas:

Annotations instead of swimlanes

Los responsables de la ejecución de una actividad se puede representar mediante una anotación colocada entre paréntesis.



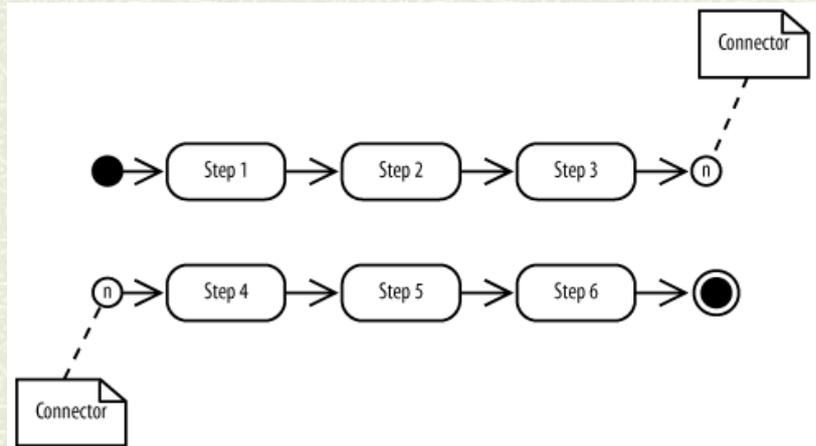
Notas:

Connectors

Cuando un diagrama de actividad es muy complejo se puede descomponer en distintos diagramas utilizando conectores.

Se representa como un círculo con un identificador en su interior.

Dos puntos del diagrama de actividad marcados con el mismo conector representan el mismo punto del diagrama.



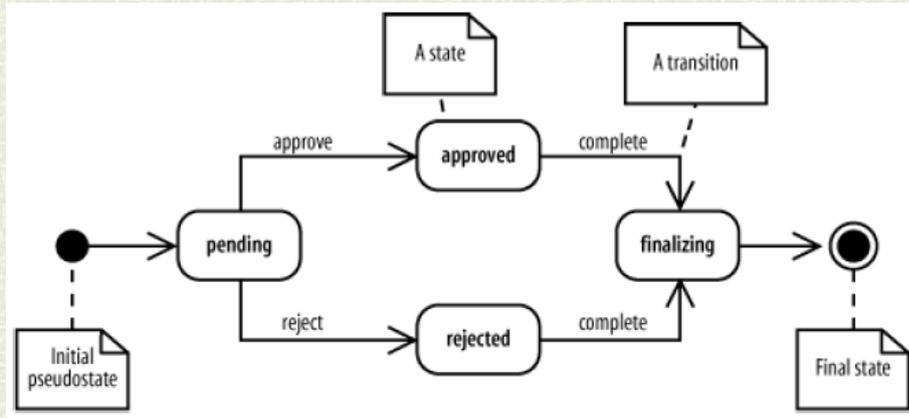
Notas:

Diagramas de máquinas de estado (diagramas de estados)

- ✦ Los diagramas de máquinas de estado son útiles para describir el comportamiento de clases y sistemas que han sido concebidos haciendo uso de un modelo de estados.
- ✦ En un modelo de estados se identifican las situaciones en la que el comportamiento o capacidad de respuesta con cualitativamente diferentes, así como los eventos o condiciones bajo las que se pasa de una situación a otra (transiciones de estados).
- ✦ Los diagramas de estados son intensivamente utilizados en:
 - Sistemas de tiempo real y críticos.
 - La descripción de sistemas reactivos.
 - La descripción de sistemas basados en protocolos

Notas:

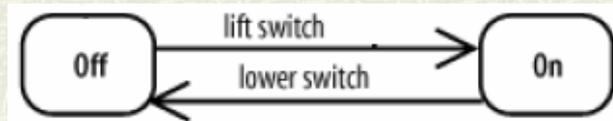
Elementos básicos de un diagrama de estados



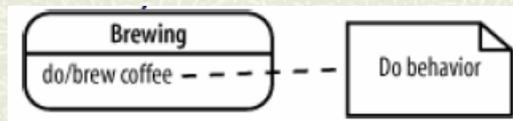
Notas:

-
- ✦ Un estado puede calificarse:
- Estáticamente en función del valor que tienen sus atributos.
 - Dinámicamente, esto es en función de la actividad que ejecuta.

Estáticamente



Dinámicamente

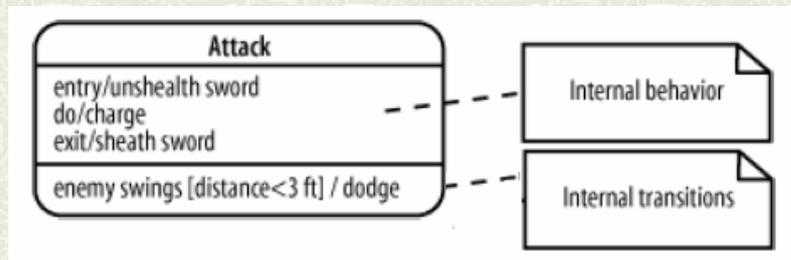


Notas:

Comportamiento completo de un estado

El comportamiento describe las acciones que se producen mientras que el sistema se encuentra en un estado:

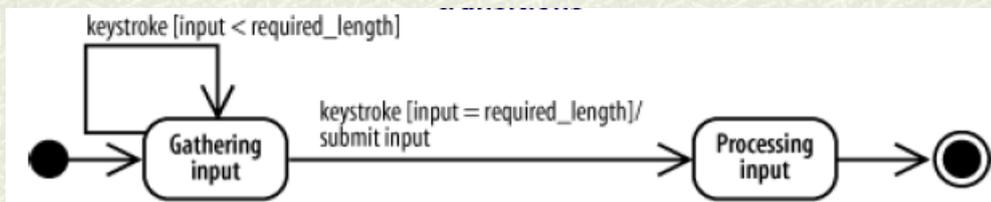
- entry/behavior => Acción que se realiza cuando se llega a un estado.
- do/behavior => Actividad que se ejecuta mientras se está en un estado.
- Exit/behavior => Acciones que se ejecuta cuando se abandona un estado.
- Transiciones internas => Se formulan como *trigger[guard]/behavior*



Notas:

Transiciones

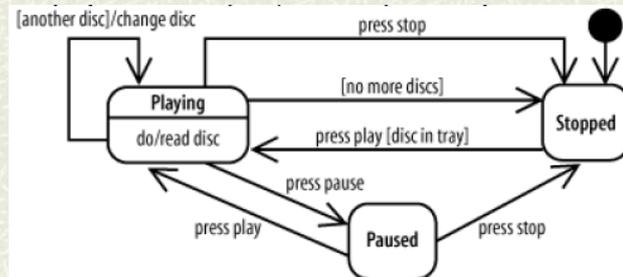
- ✦ Una transición representa las causas circunstancias y efectos de un cambio entre dos estados.
- ✦ Una transición tiene un nombre y una descripción completa del tipo *trigger[guard]/behavior* :
 - Trigger: Es el evento que da origen a una transición
 - Guard: Es una función booleana que se evalúa cuando ocurre el trigger. Si es True la transición se produce. Si vale False la transición no ocurre.
 - Behavior: Es una acción ininterrumpible que tiene lugar cuando se produce la transición.



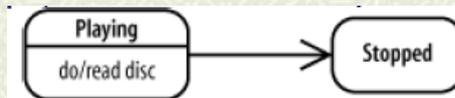
Notas:

Múltiples transiciones

- La posibles combinación de triggers y condiciones de guardas dan lugar a diferentes conjuntos de transiciones entre estados:



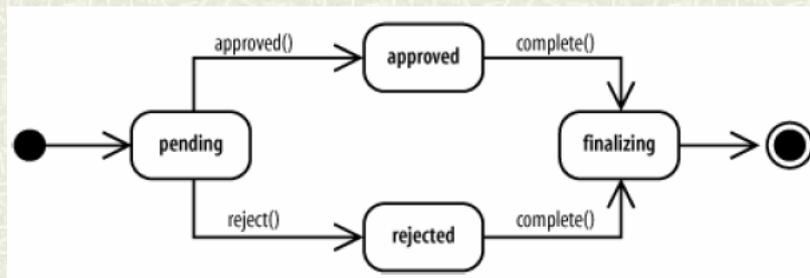
- Si una transición no tiene trigger ni guarda, la transición se produce por finalización de la actividad asociada al estado.



Notas:

Trigger, guardas y respuesta en programas

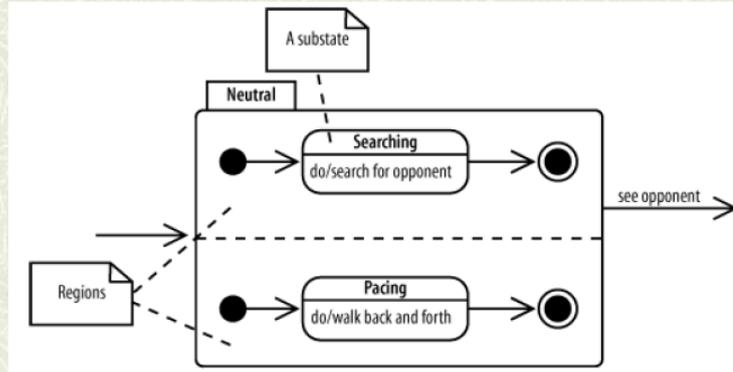
- ✦ Es muy frecuente cuando se modela software que:
 - El trigger puede ser la invocación de un metodo de la clase modelada.
 - La guarda es la invocación de un metodo booleano o una relación entre los atributos.
 - La respuesta puede ser la invocación de un metodo sobre el propio objeto o sobre un objeto al que tenga acceso.



Notas:

Estados compuestos

- UML permite que varias máquinas de estados se asocien a una misma clase o sistema. Su situación puede estar descrita en un mismo instante por varios estados, cada uno de ellos pertenecientes a diferentes máquinas.



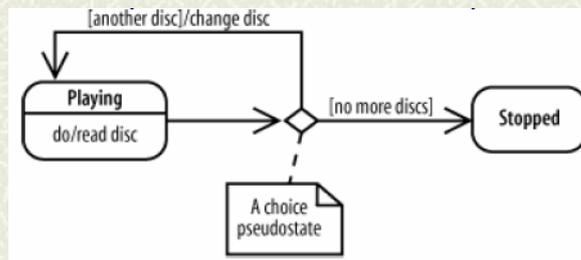
- Un estado de inicio histórico asociado a una región de estado representa que cuando se accede a la región se accede al estado en que se encontraba la región cuando se abandonó.



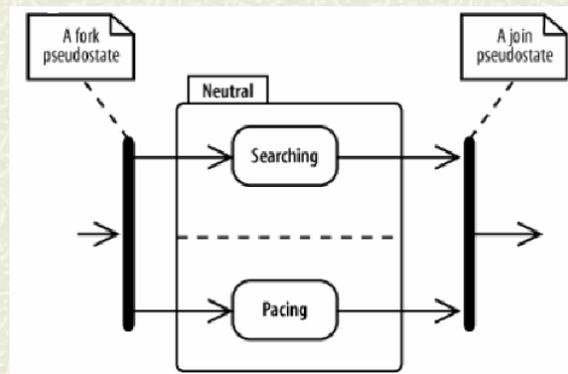
Notas:

Pseudoestados

Choice

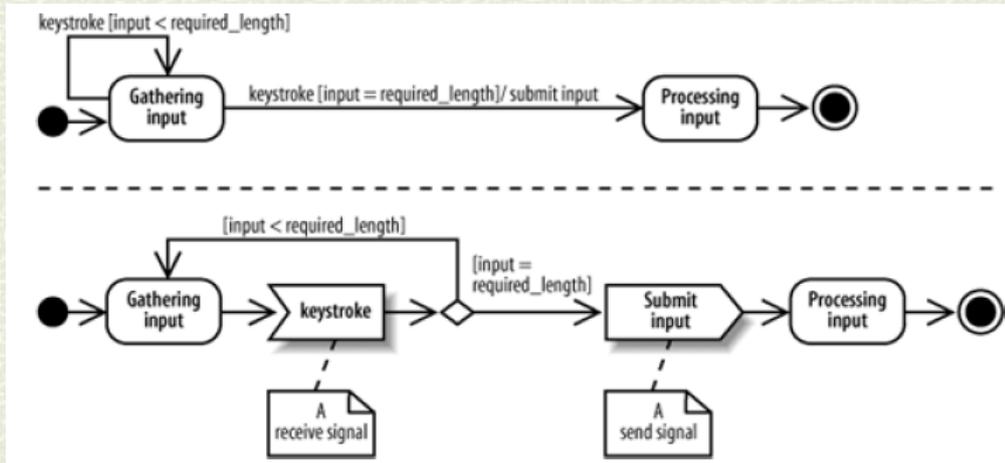


Fork /Join

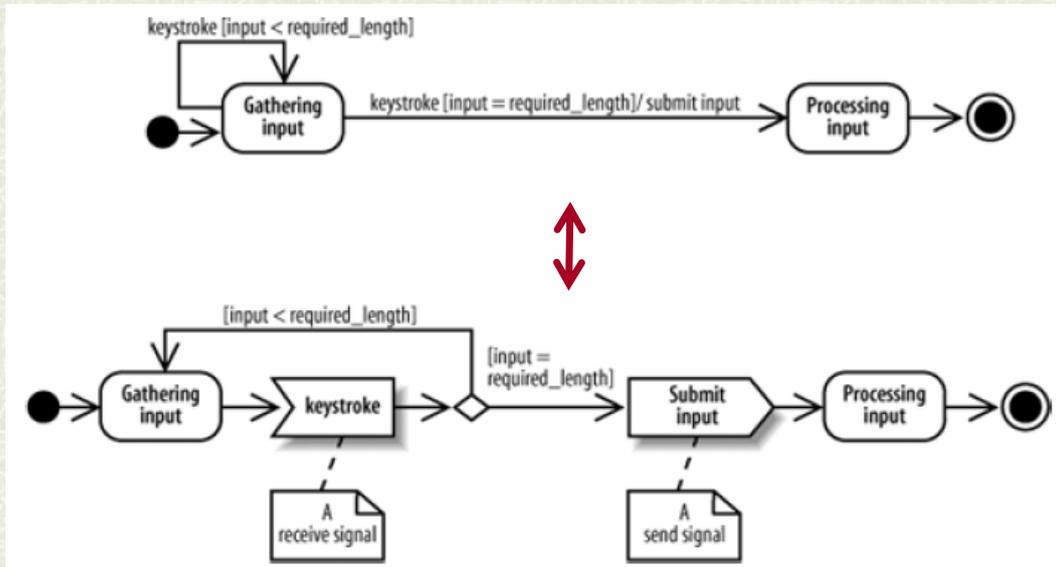


Notas:

Recepción y transmisión de señales



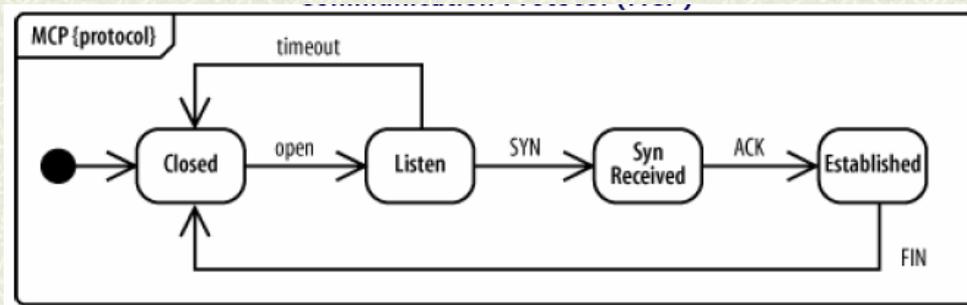
Notas:



Notas:

Maquinas de estados de protocolo

- # Una máquina de estados de protocolos es un tipo especial de máquina de estados que se utiliza para describir un protocolo de interacción.
- # No describe el comportamiento asociado a los estados y a las transiciones.
- # Resalta las secuencias de estados que son posibles y los eventos que dan lugar a esas transiciones.
- # Se representa en una caja rectangular con una pestaña que contiene el esterotipo <<protocol>>



Notas: