

UML

Modelado Dinámico

Diagramas de Actividades y Diagramas de Secuencia

Basado en OMG Unified Modeling Language™ (OMG UML). Version 2.5.

Definición general

Una actividad es una clase de comportamiento especificado por medio de una gráfica con nodos interconectados.

Un subconjunto de estos son ***nodos ejecutables(Acciones)*** e incluyen pasos discretos de ejecución. Otro subconjunto se corresponde con ***nodos de control*** los cuales determinan la secuencia del flujo de ejecución.

Nodos objeto cargan datos que pueden entrar y salir de los ***nodos ejecutables***.

Por consiguiente, los diagramas de actividad son lo que se ha denominado comúnmente ***modelos de flujo de datos y control***

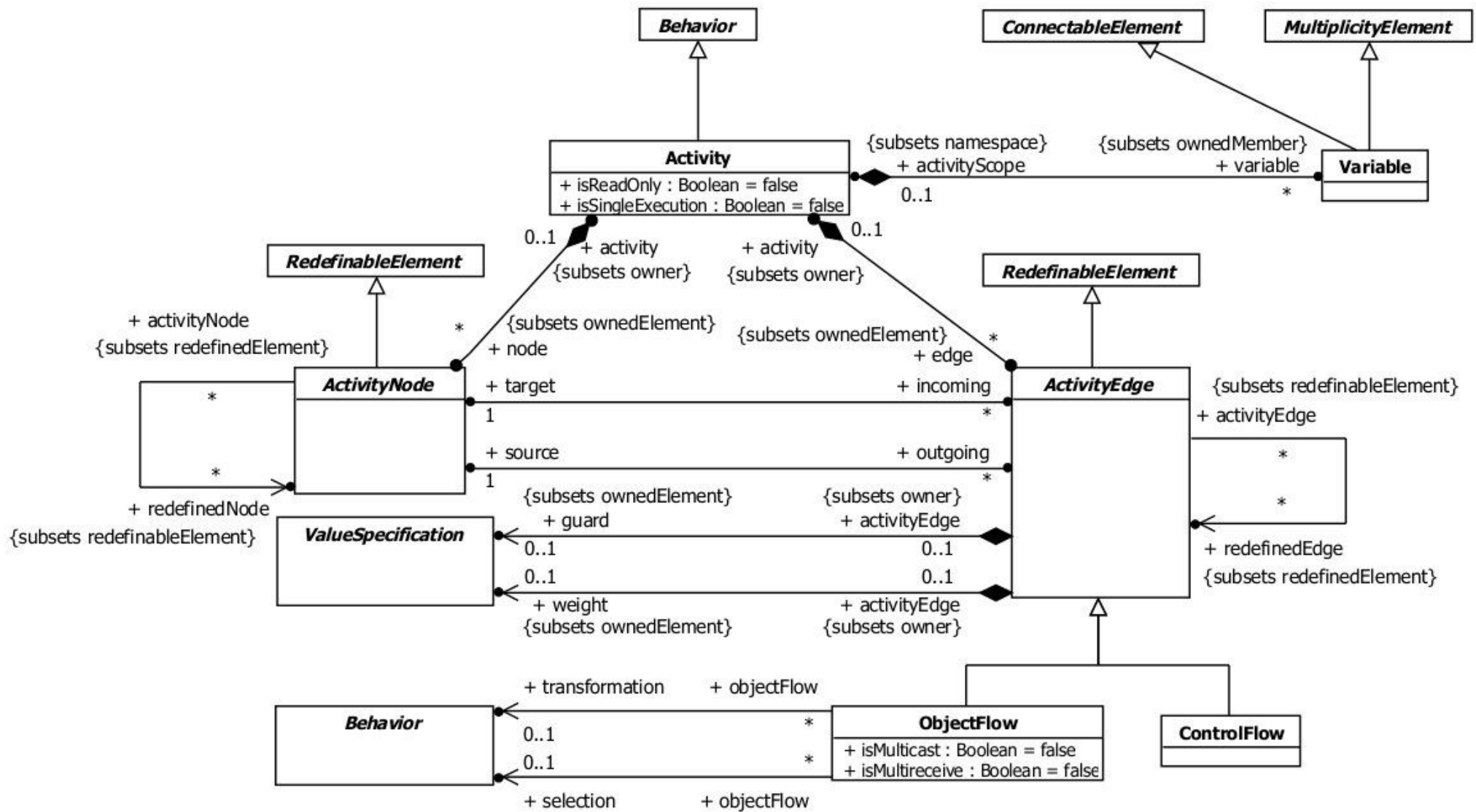
Diagramas de Actividades

Se corresponde con los denominados **flujos de control** y modelos de **flujo de objetos**

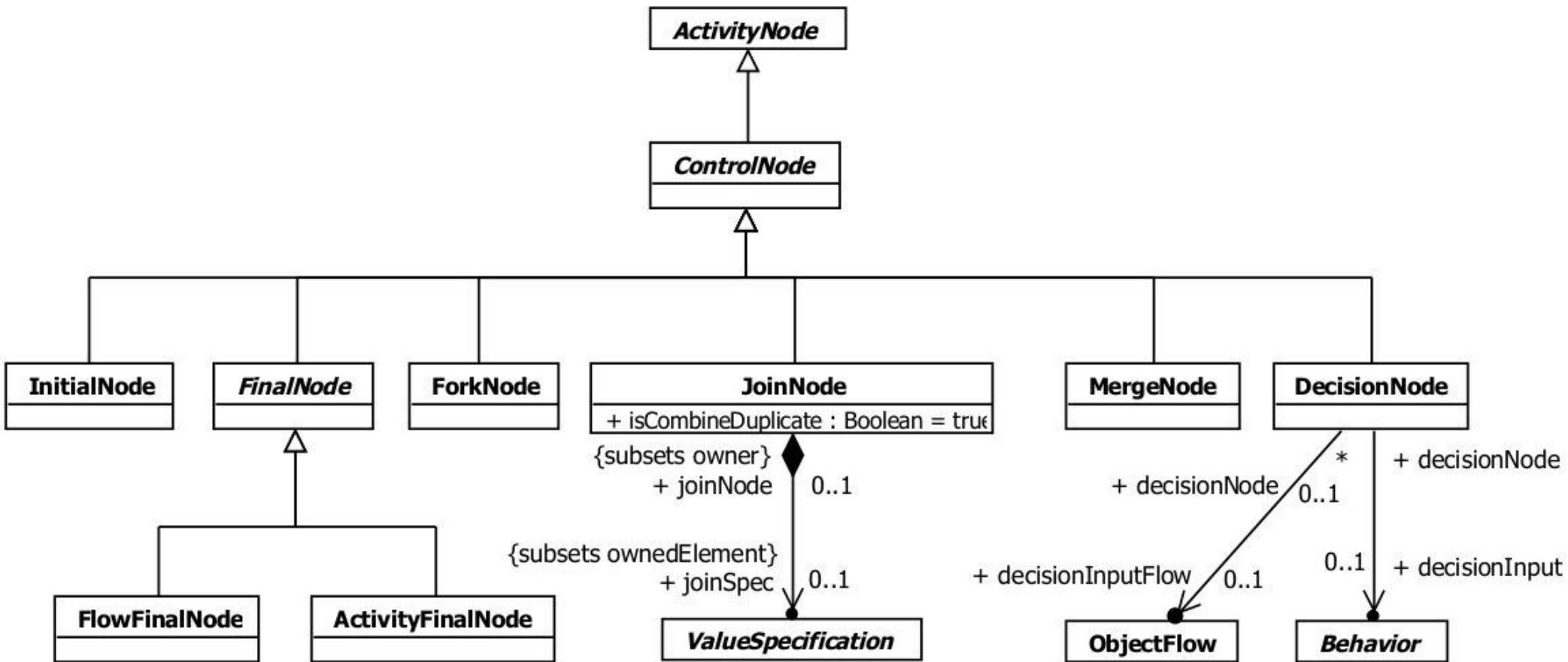
Niveles de especificación

- **Nivel básico.** Diagramas de flujo secuenciales tradicionales.
- 2. **Nivel intermedio.** Actividades concurrentes y flujo de datos.
- 3. **Nivel completo.** Ponderaciones de flujos y “streaming”.
- 4. **Nivel estructurado.** Programación estructurada convencional(ciclos, condicionales). Compatible con nivel completo e intermedio.
- 5. **Nivel estructurado completo.** “Pines” de salida de flujo de datos en ciclos y condicionales. Orientado a ser usado con el nivel intermedio para soportar concurrencia explícita aunque son ortogonales.
- 6. **Nivel extraestructurado.** Manejo de excepciones como en los lenguajes de programación tradicionales e invocación de comportamientos sobre conjuntos de valores. Requiere el nivel estructurado

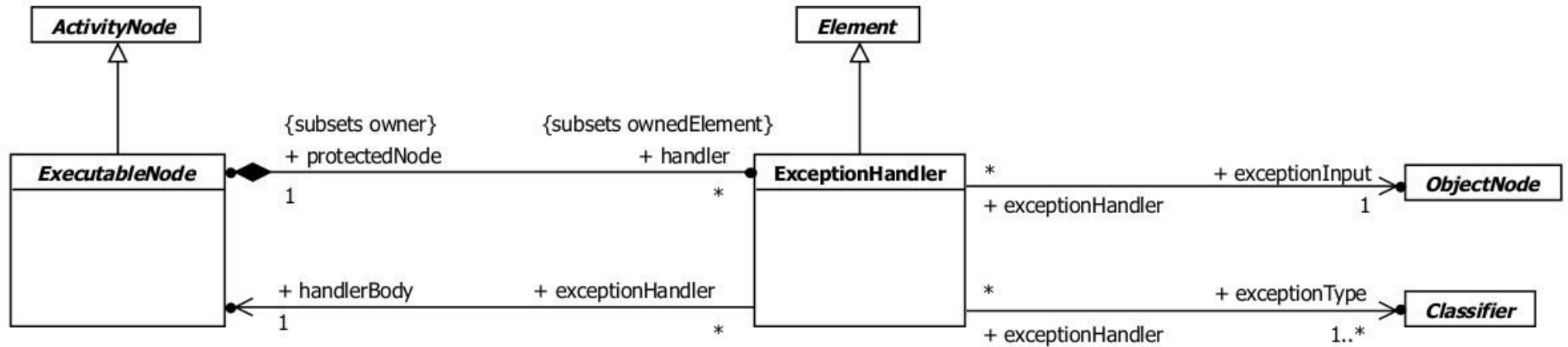
Sintaxis abstracta de actividades



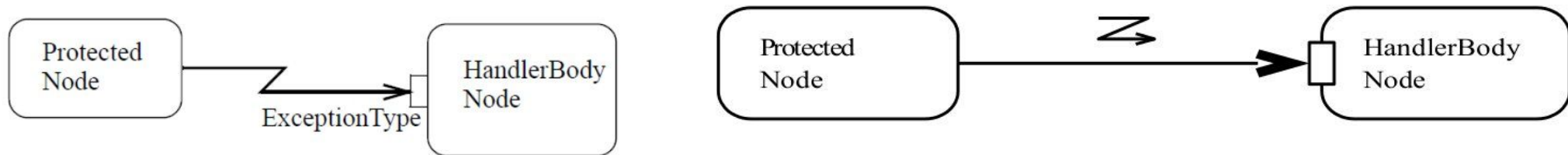
Sintaxis abstracta de nodos de control



Sintaxis abstracta nodo ejecutable



Ejemplos de representación de manejo de excepciones



Resumen de nodos de actividad



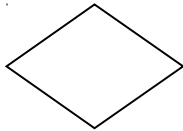
Nodo ejecutable (Acción)



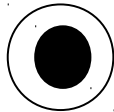
Nodo objeto



Final flujo



**Nodo de decisión
O nodo “merge”**



Final actividad



Inicio actividad

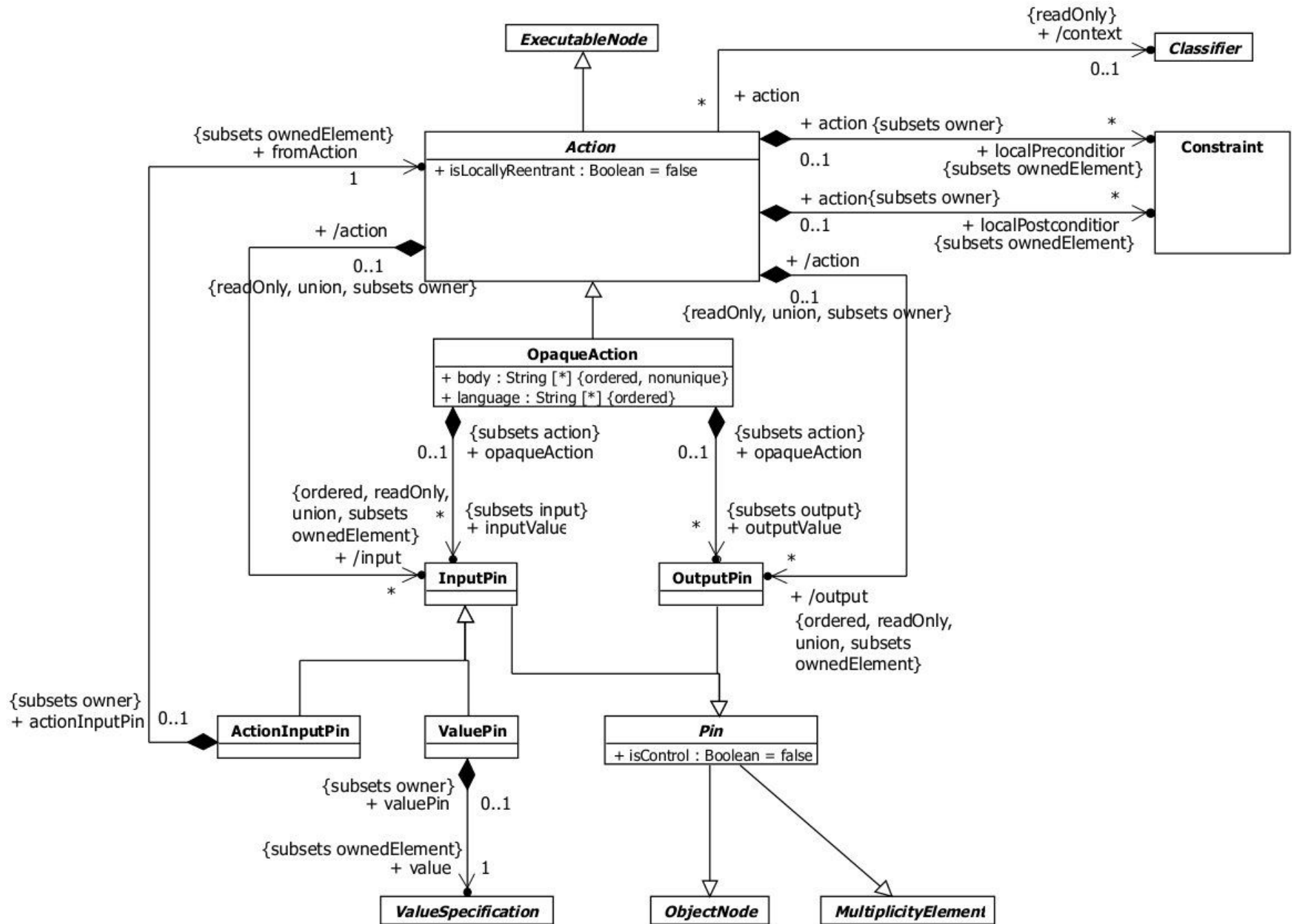


“fork” o “join”

Nodos de control

Preparó HAD.

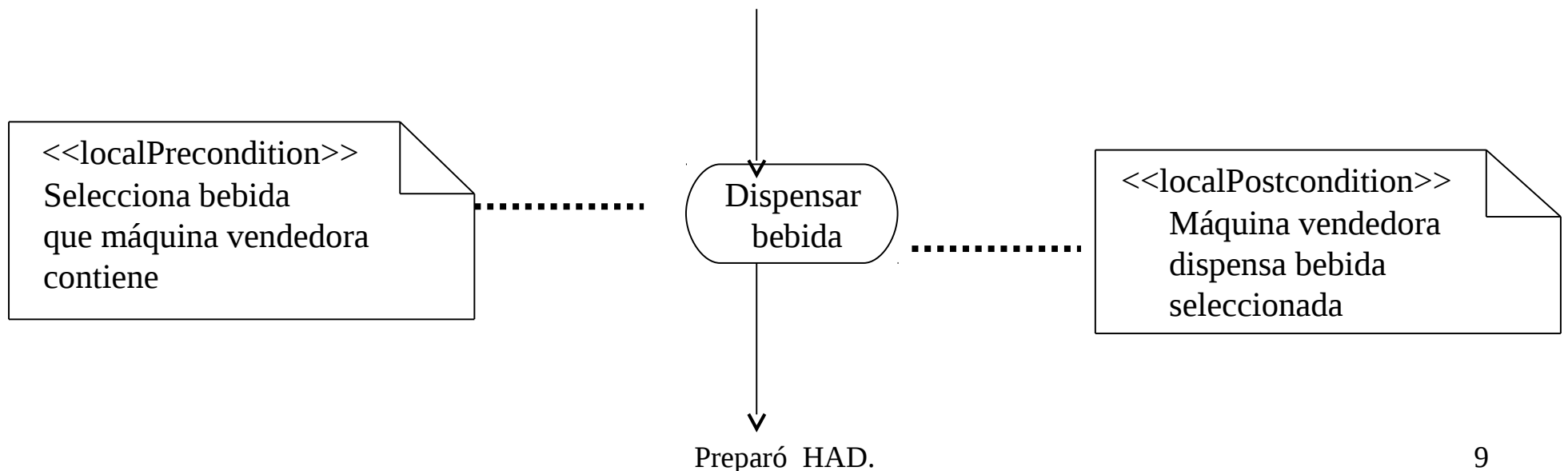
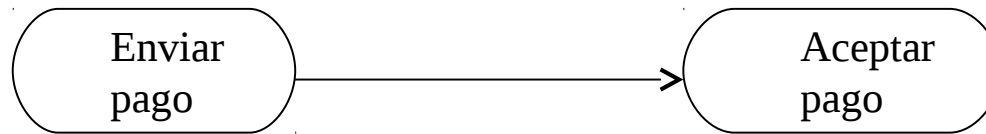
Sintaxis abstracta de acción



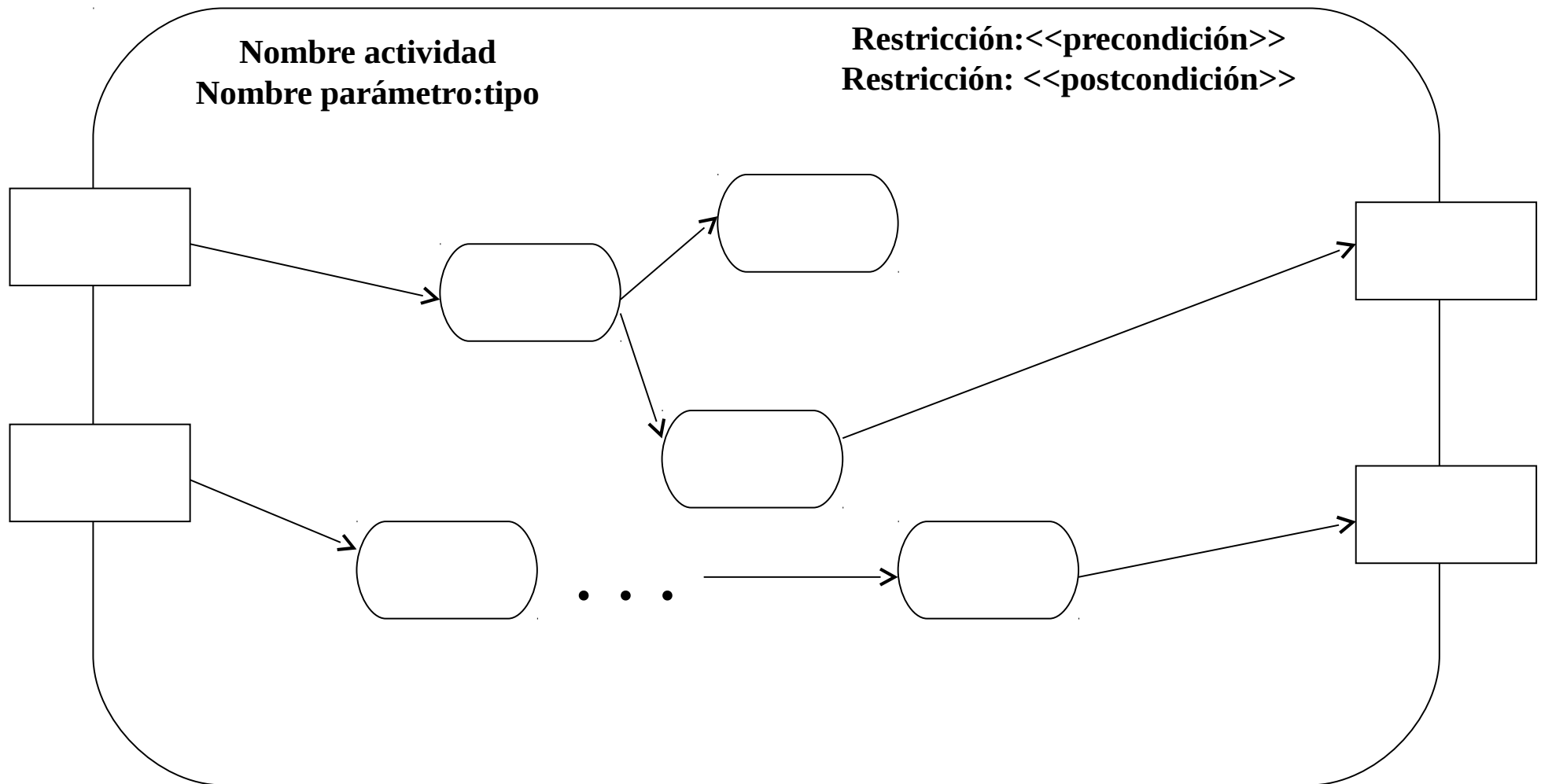
Acción

Es un nodo ejecutable como unidad fundamental de funcionalidad en una actividad, diferenciándose así del flujo de control y de datos entre Acciones.

NOTACIÓN

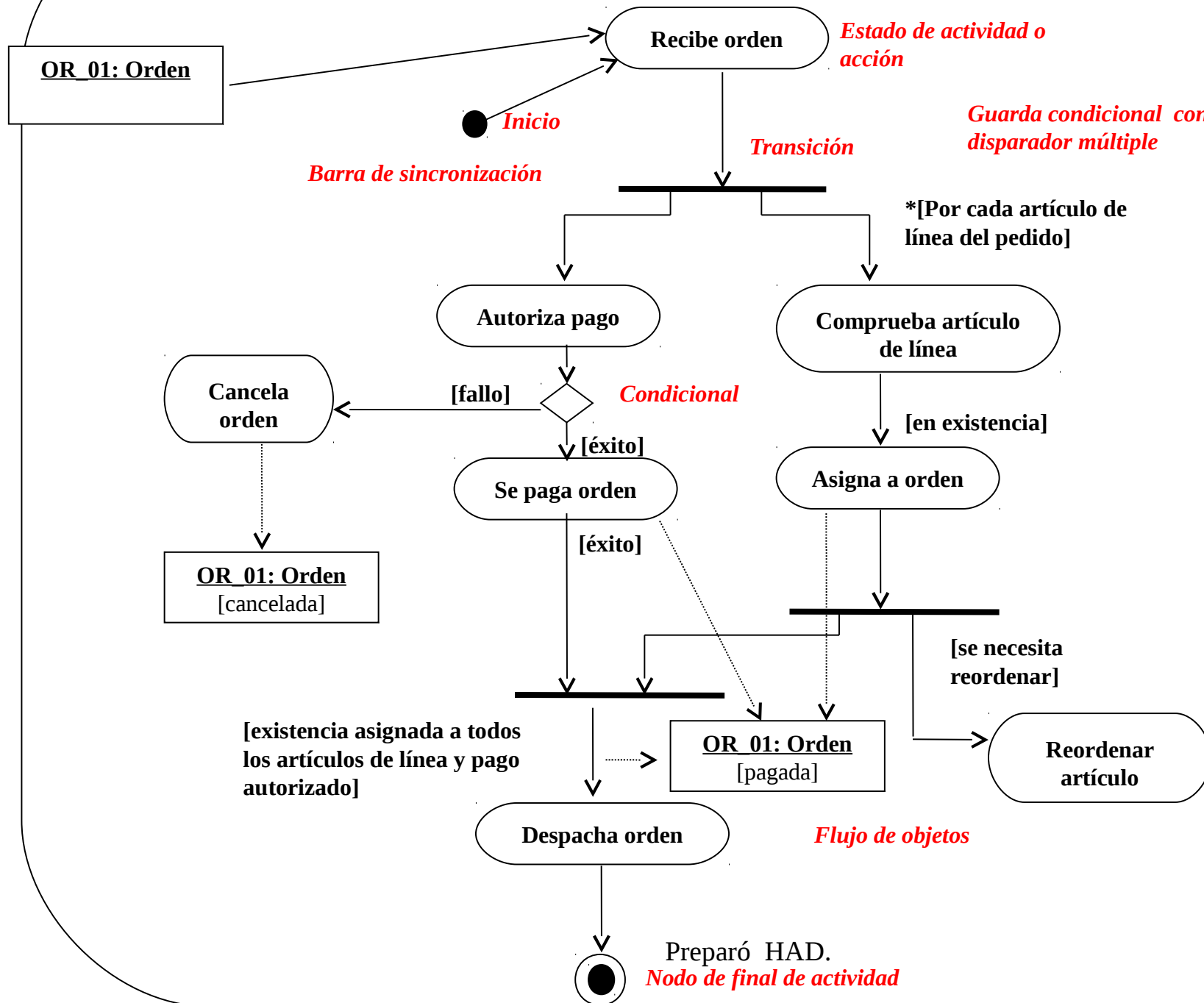


Actividad(I)

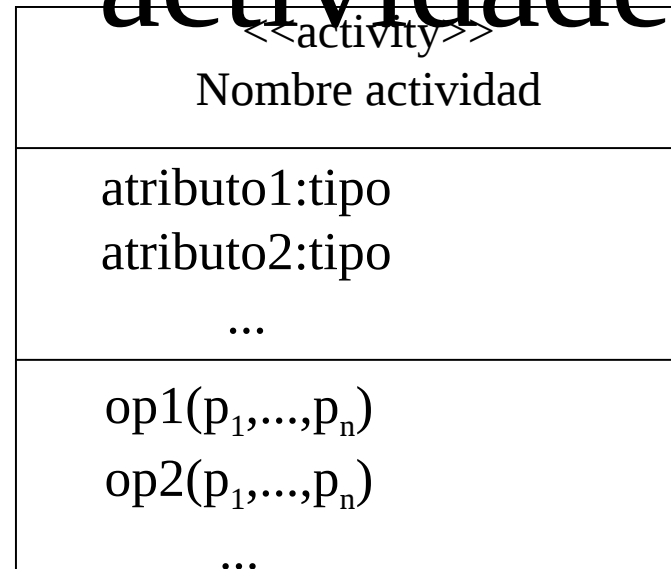


Procesar pedido OR 01:Orden

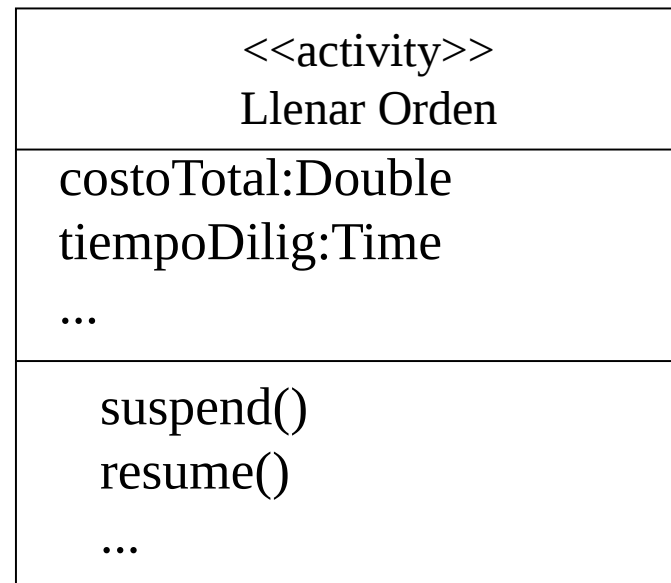
<<precondición>>Orden completa
<<postcondición>>Orden cerrada



Notación de clase para actividades

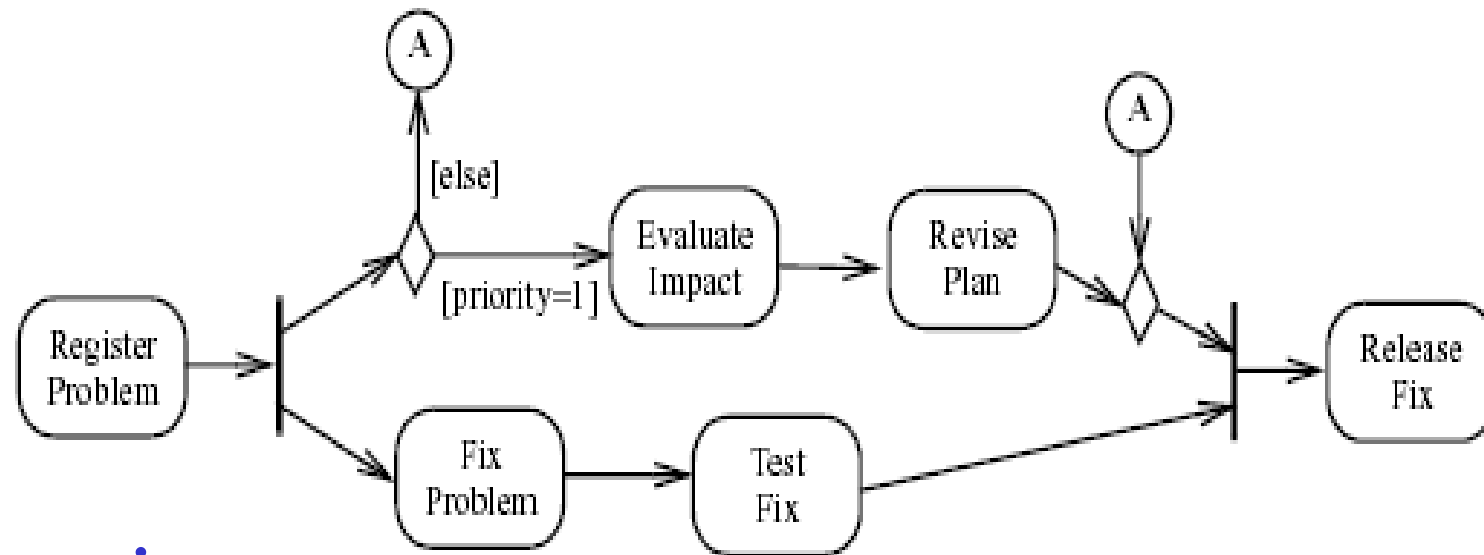


Ejemplo:

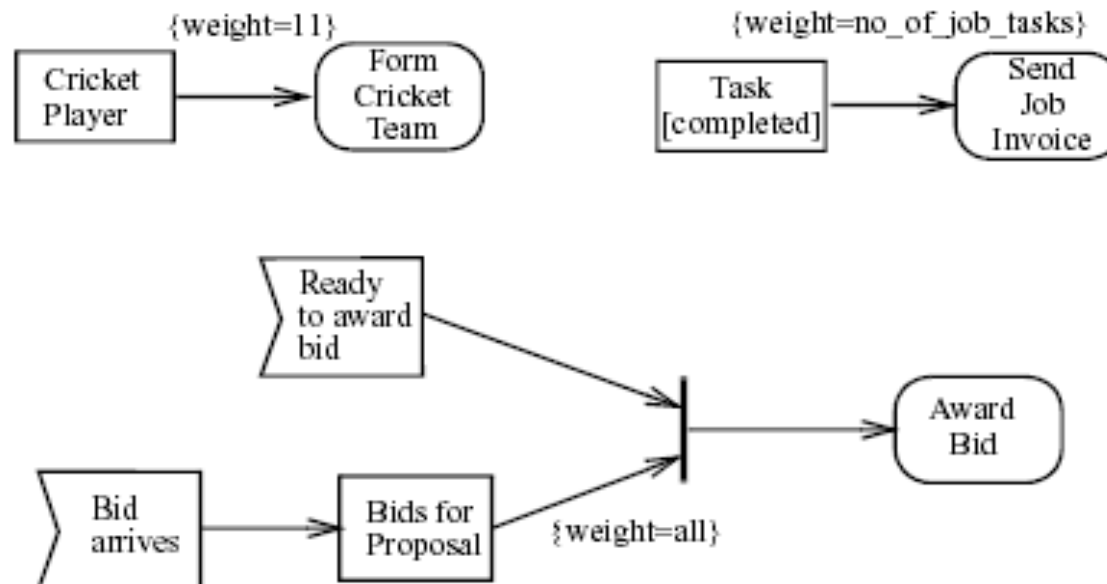


Flujos de actividad(Bordes)

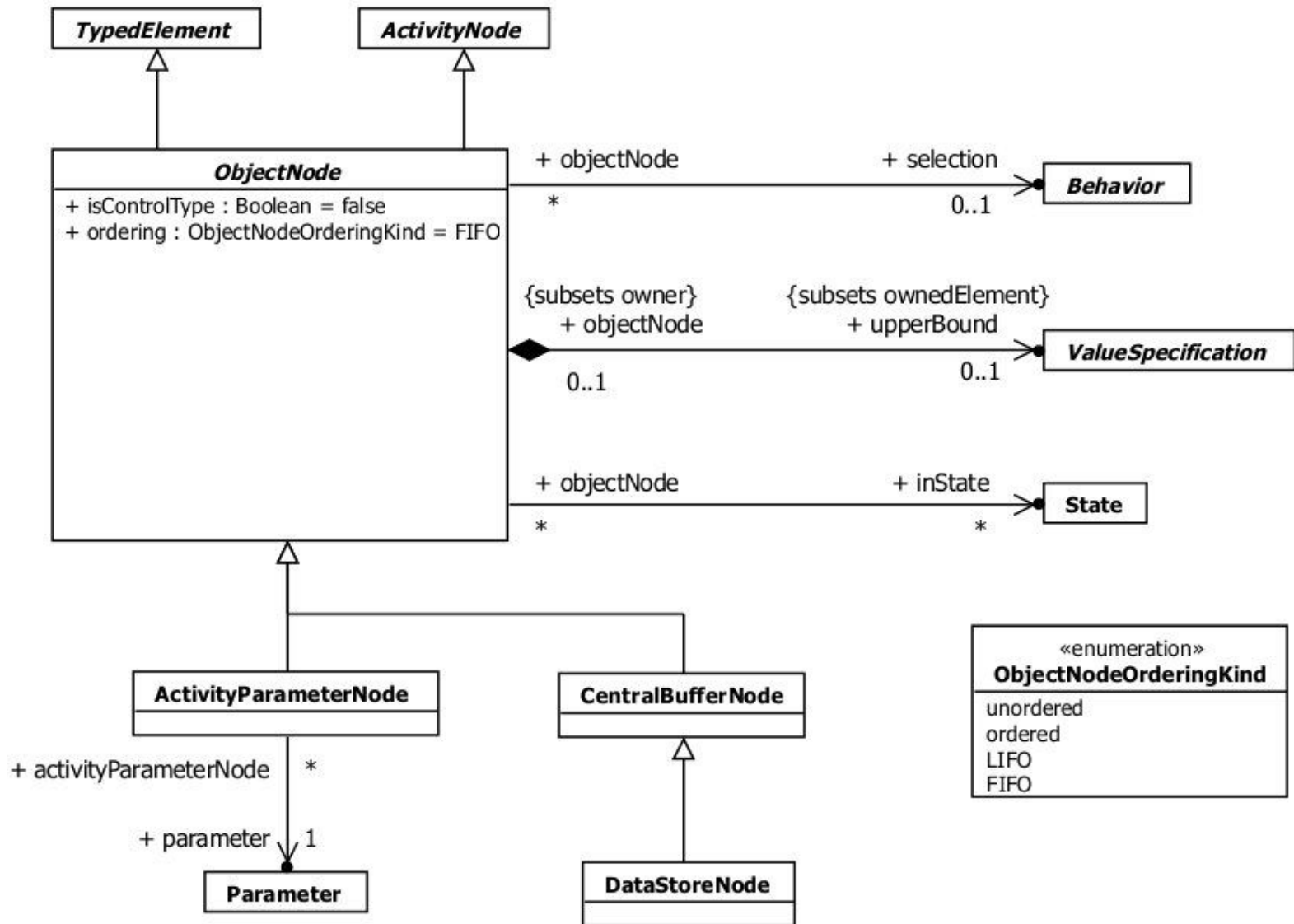
Conectores



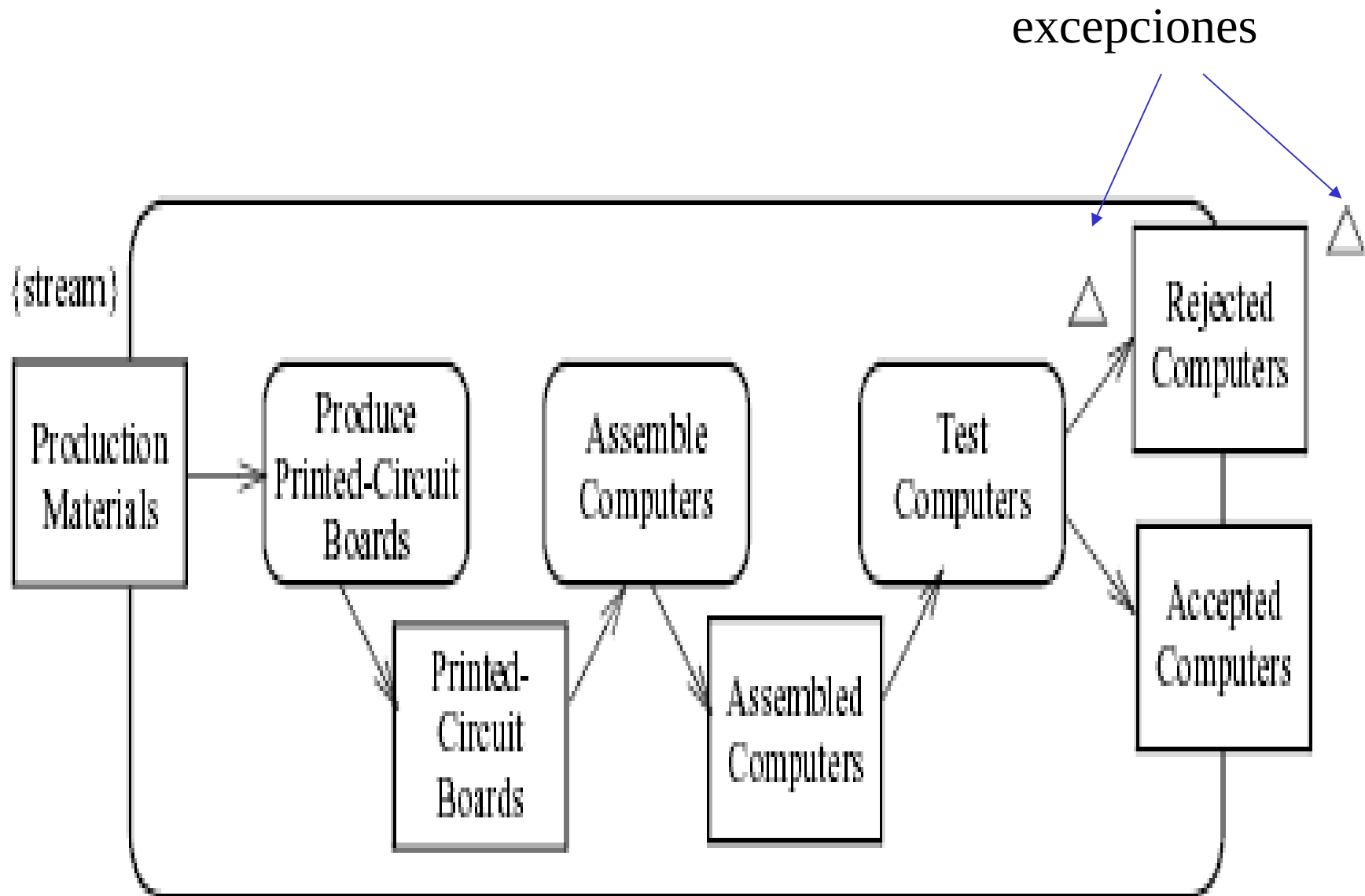
Ponderaciones



Sintaxis abstracta nodos objeto

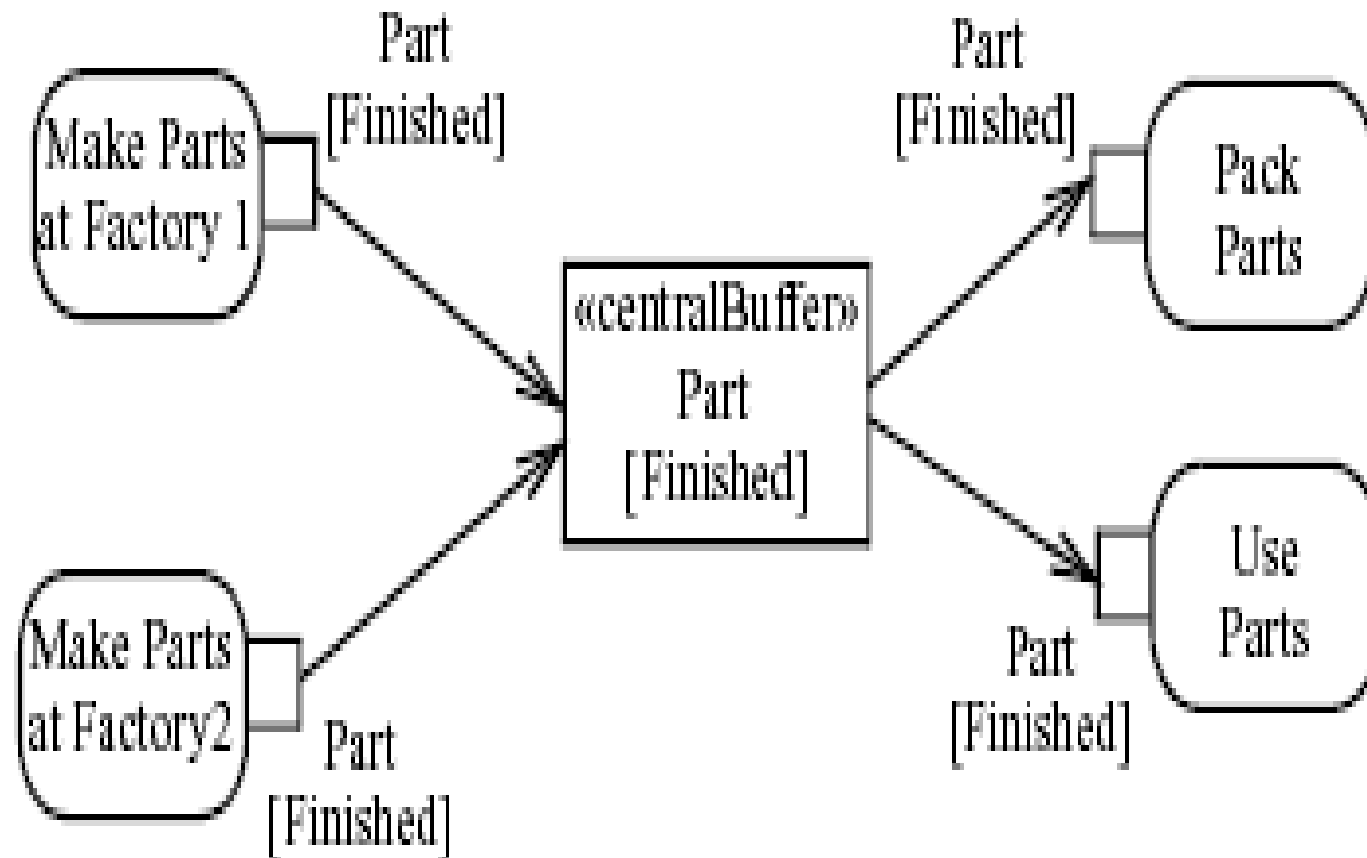


Nodos parámetro



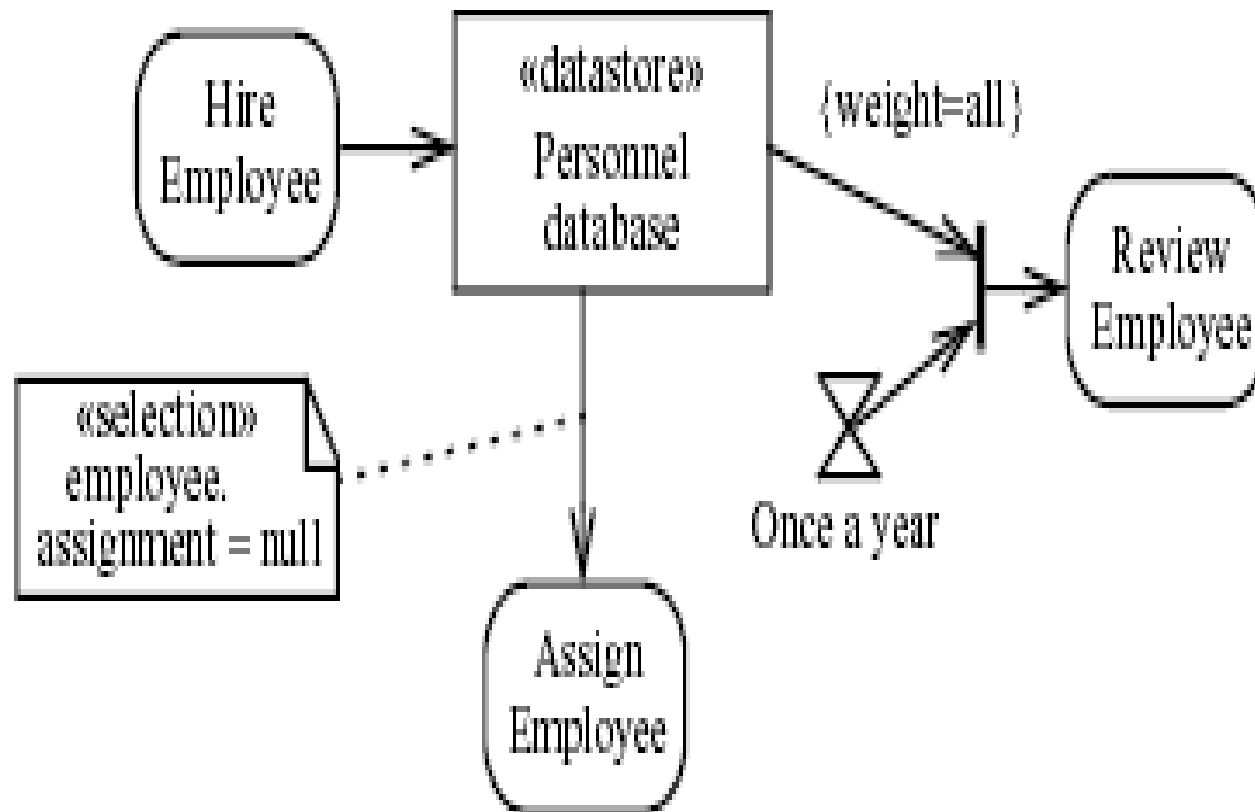
“CentralBufferNode”

Es un nodo objeto para manejar flujos desde múltiples fuentes y destinos



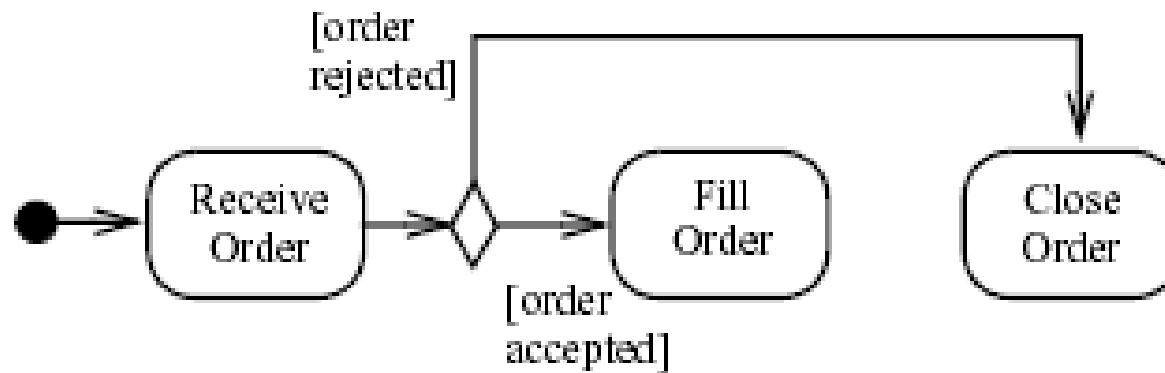
“DataStoreNode”

Es un nodo de almacenamiento para información no-temporal

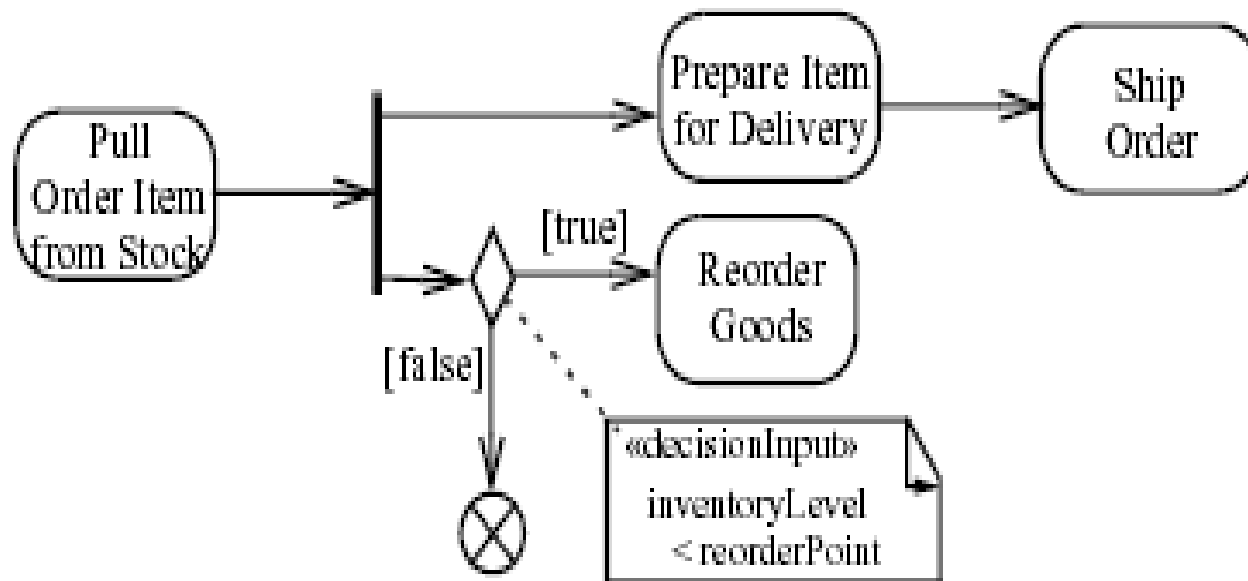


Nodos de decisión

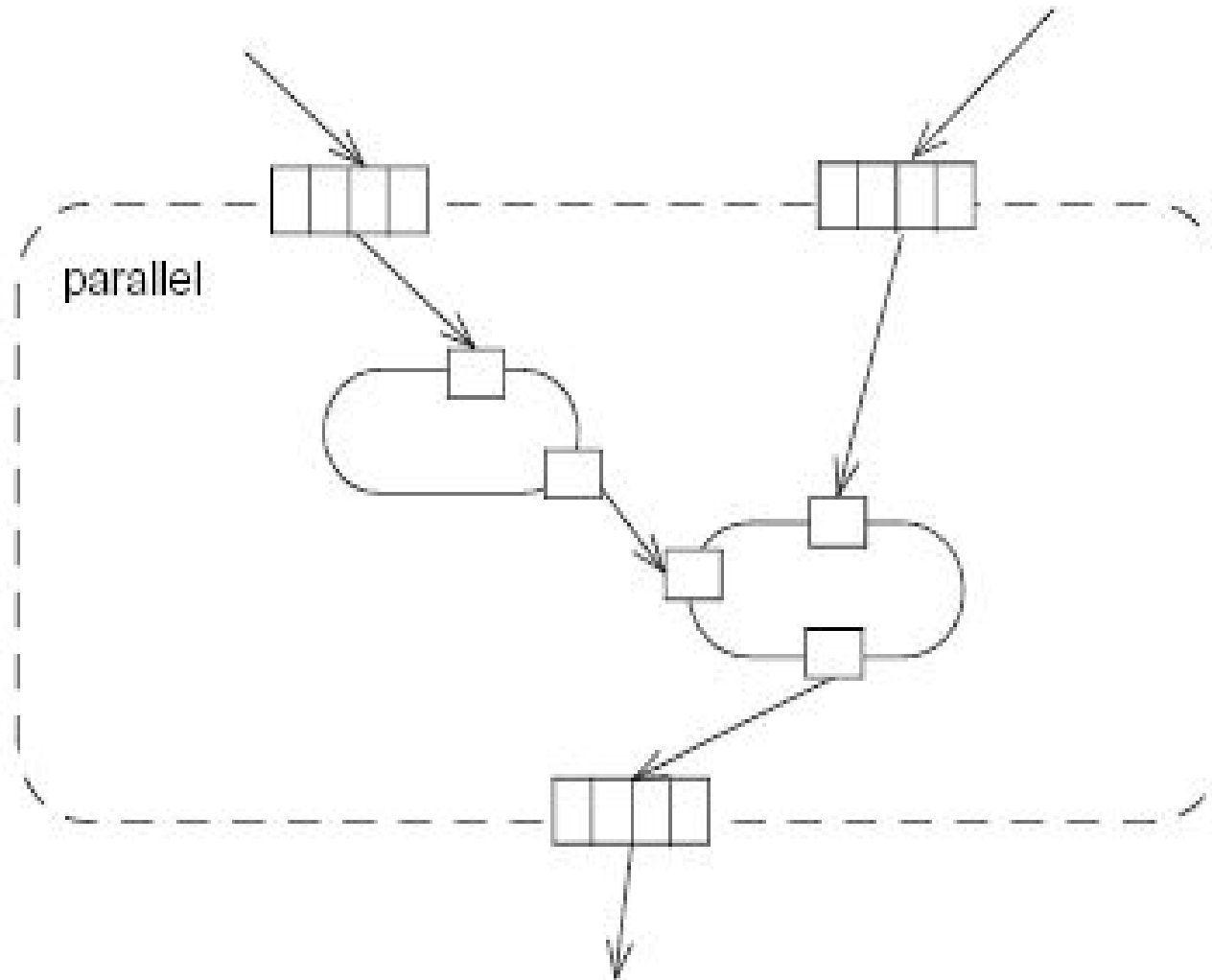
Ejemplo 1



Ejemplo 2. N.D. con comportamiento

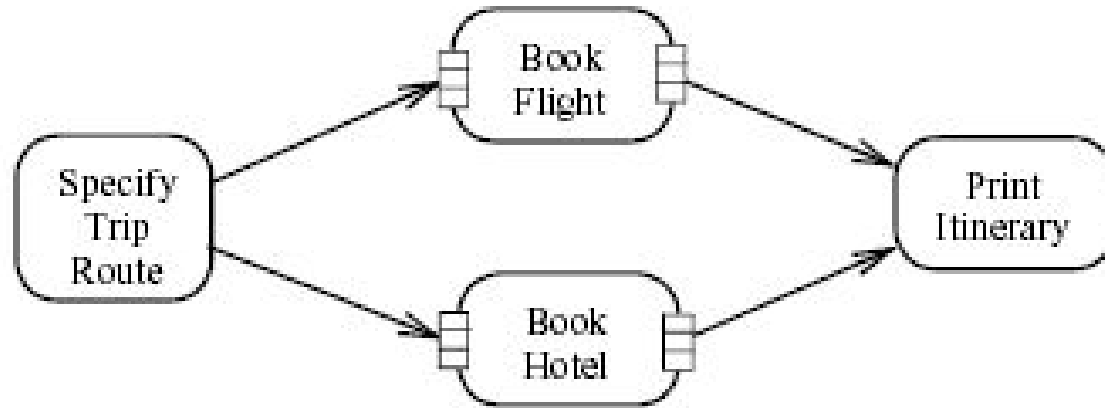


Región de expansión(I)

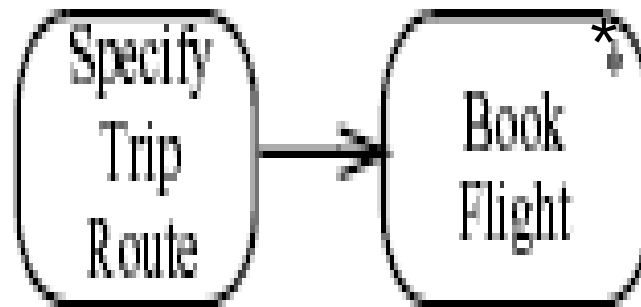


Región de expansión(II)

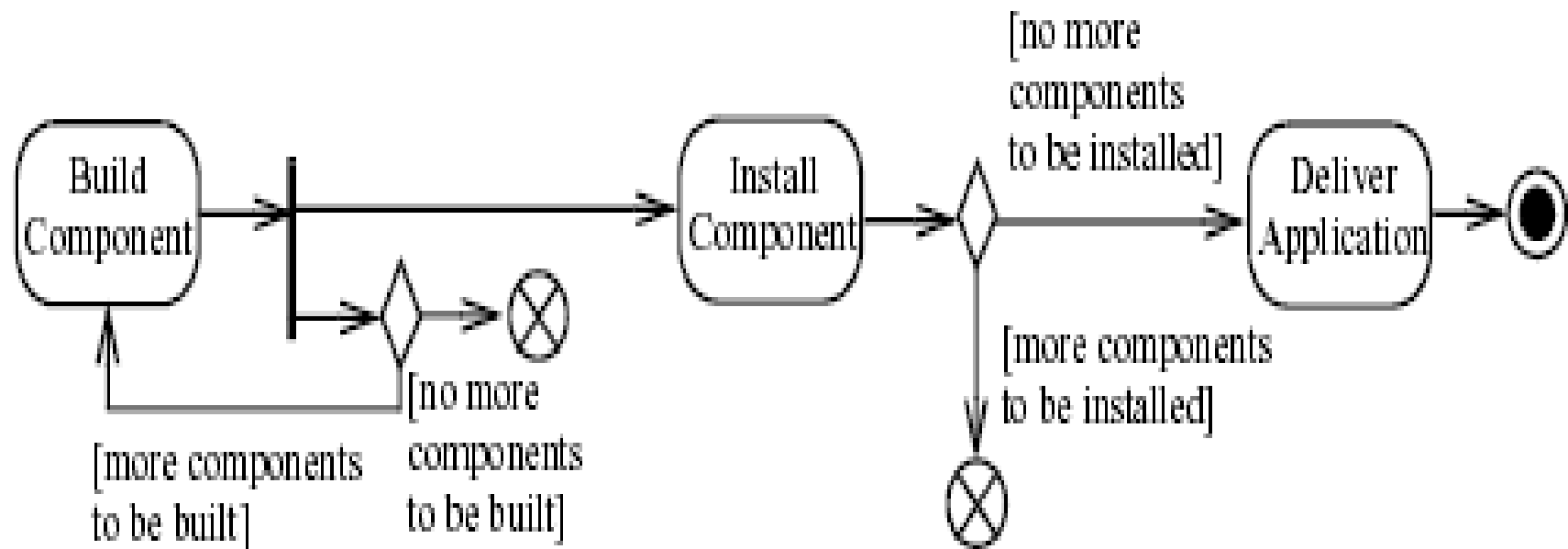
Ejemplo



Notación abreviada para la región de expansión(UML 1.x)



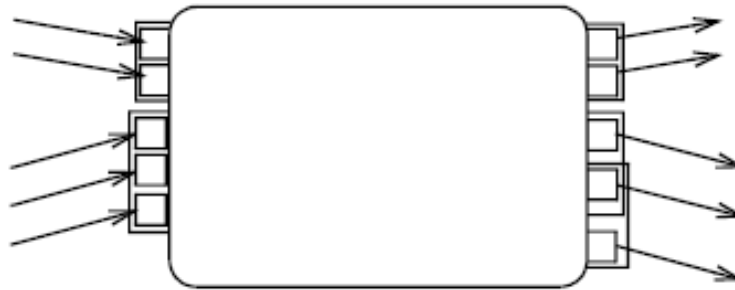
Nodos finales



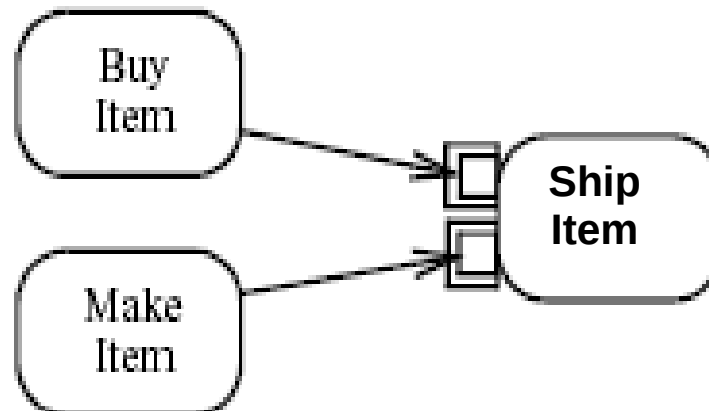
Conjuntos de parámetros

Es un elemento que permite contar con un conjunto alternativo de entradas y salidas que una actividad o entidad comportamental puede usar

Notación



Ejemplo de invocación de un “or”

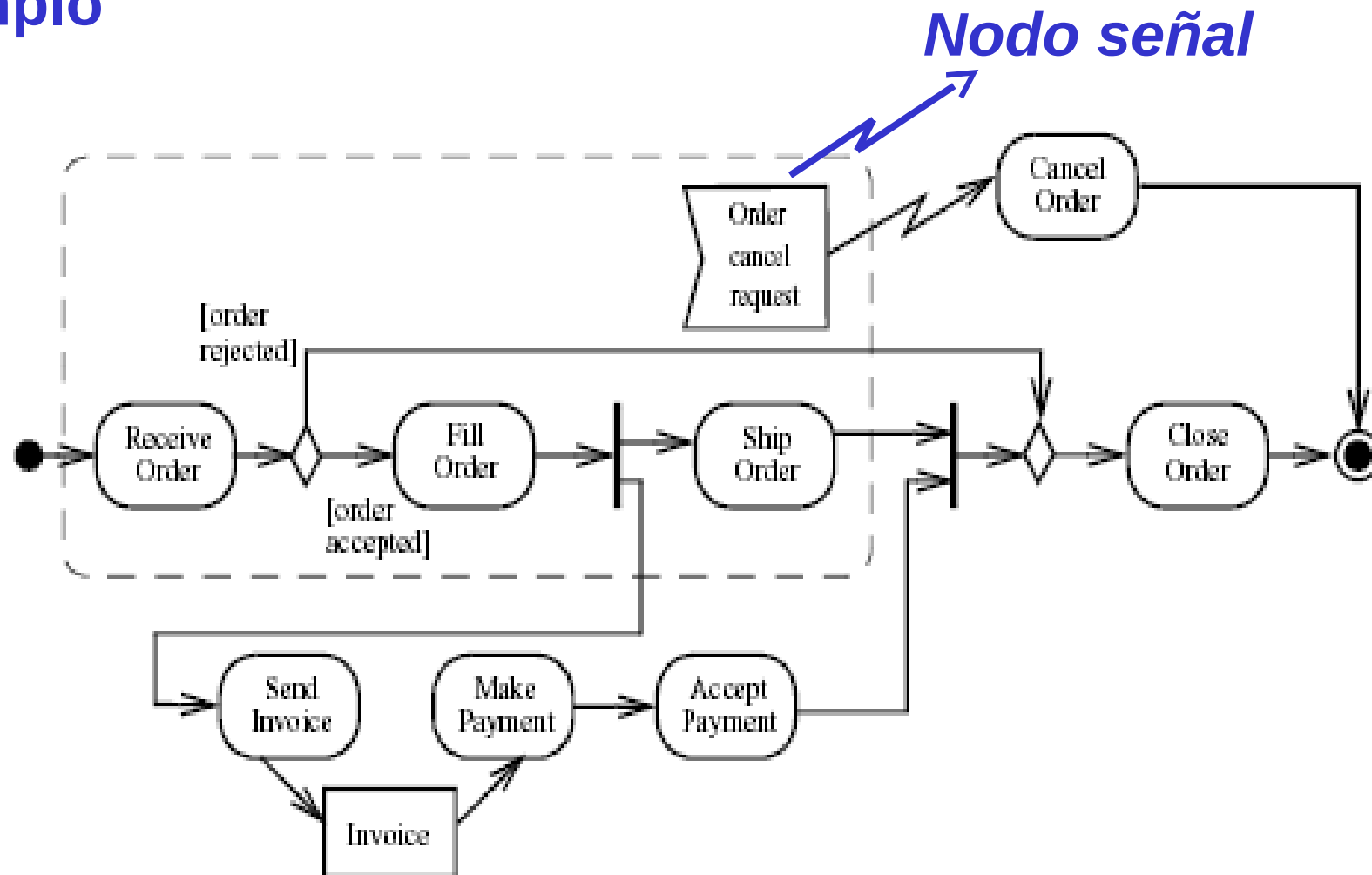


Región de actividad interrumpible

Notación

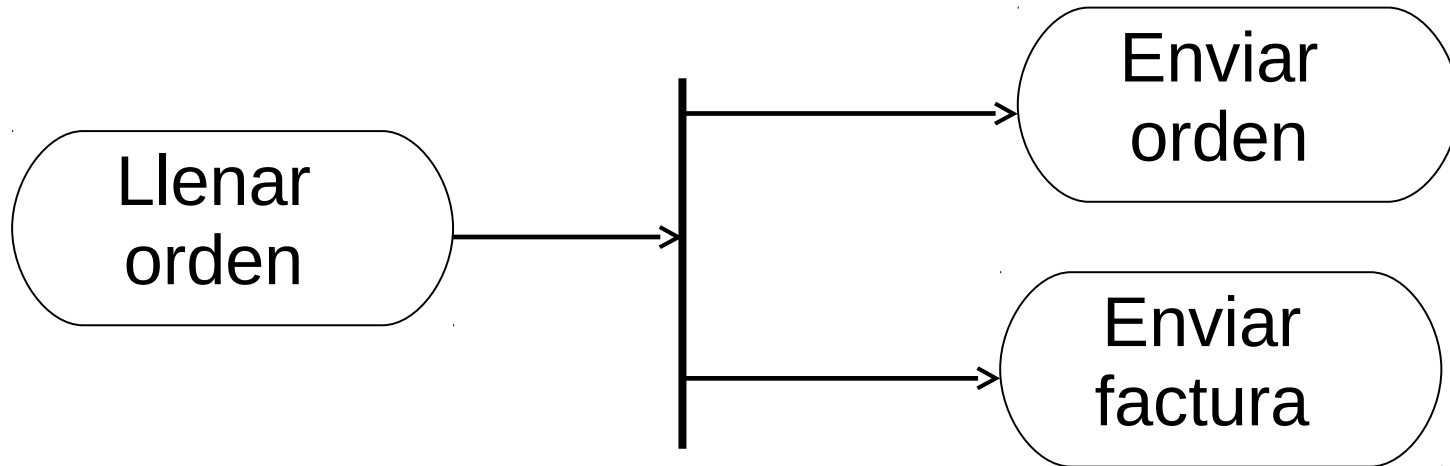


Ejemplo

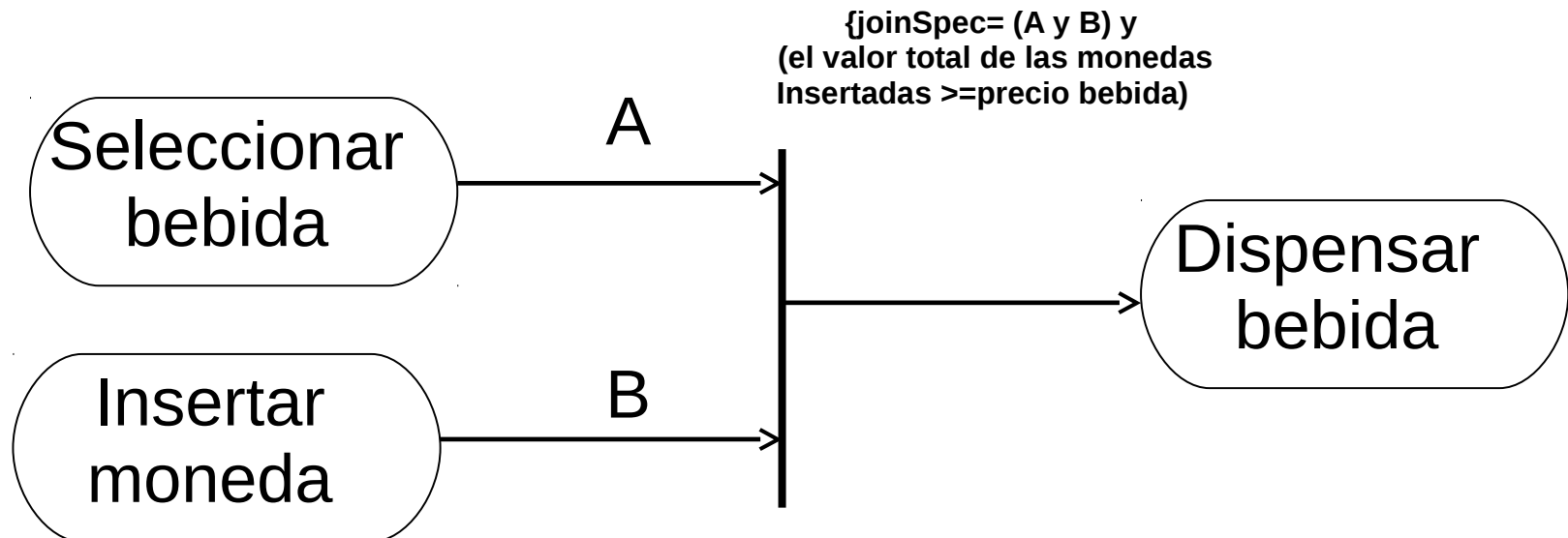


Nodos “join” y “fork”

“fork”



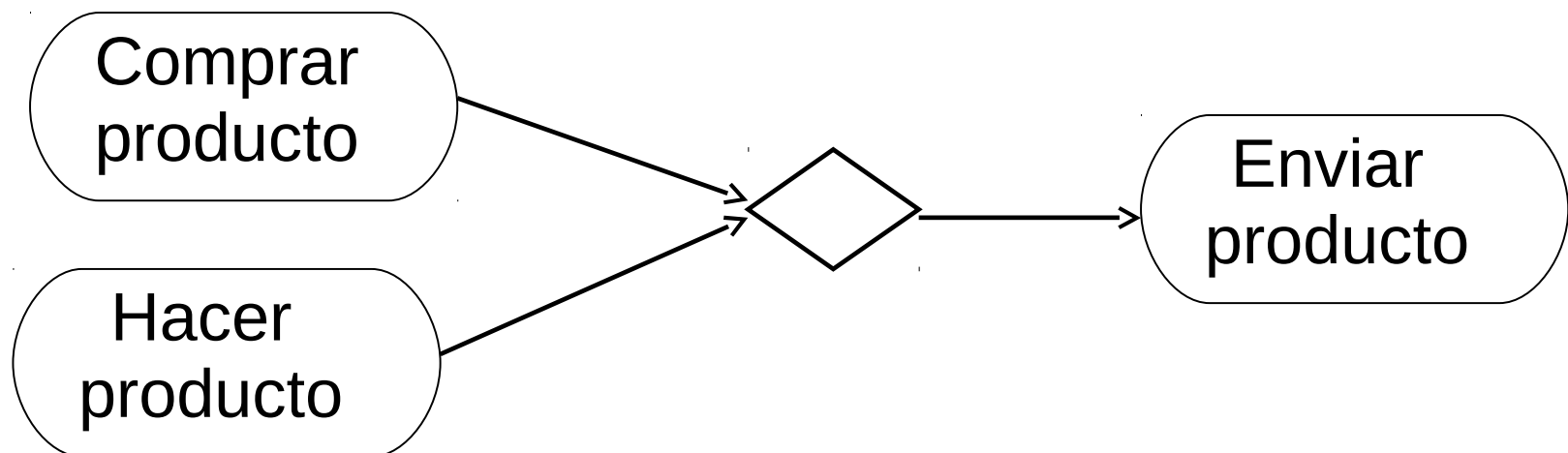
“join”



Nodo “merge”

Es un nodo de control que pone juntos múltiples flujos alternos. No se usa para sincronizar flujos concurrentes sino para aceptar uno entre diversos flujos alternos

Ejemplo:

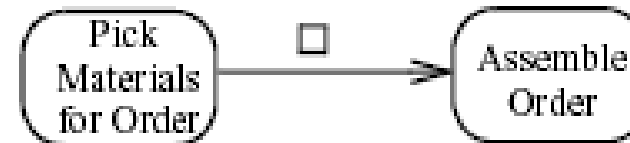
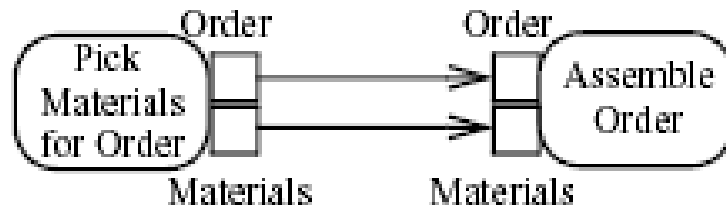


Flujos

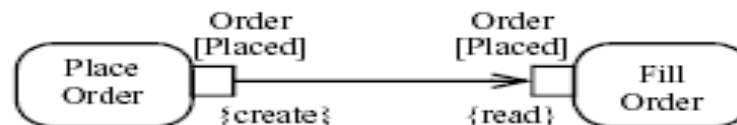
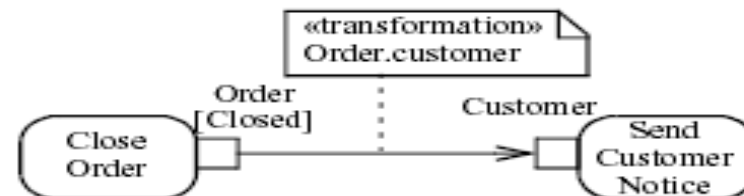
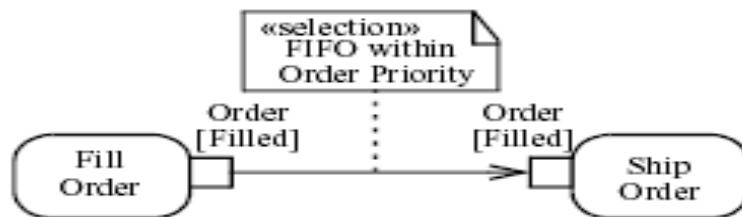
Flujos con “pines”



“Pines” explícitos y eludidos



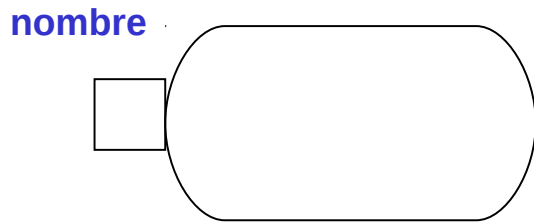
Selección comportamental en el flujo



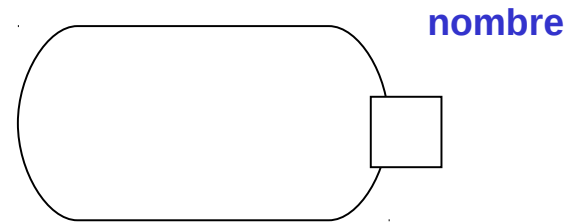
Pin(I)

un nodo objeto representando entradas y salidas desde acciones

Notación

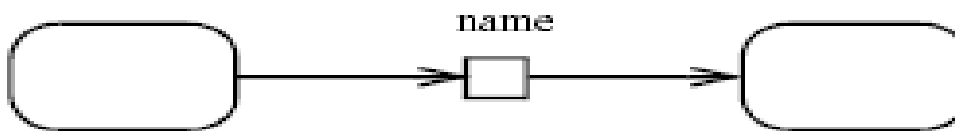


Pin de entrada



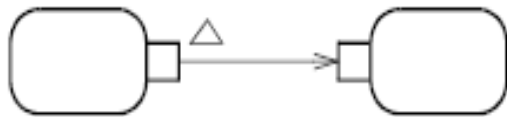
Pin de salida

Notación “pin standalone”



Pin(II)

Notación pin de excepción



Pin de salida, excepción

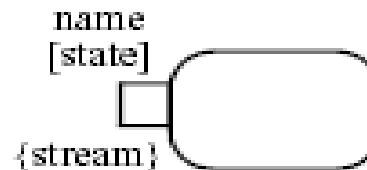


Pin de entrada y salida, estilo “standalone”, excepción

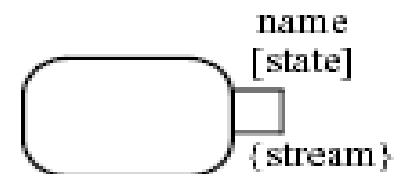
Notación pines de flujo(streaming)



*Standalone object node,,
streaming on both end*



*Input pin,
streaming*



*Output pin,
streaming*



*Input and output pin,
stand-alone style, streaming on both ends*

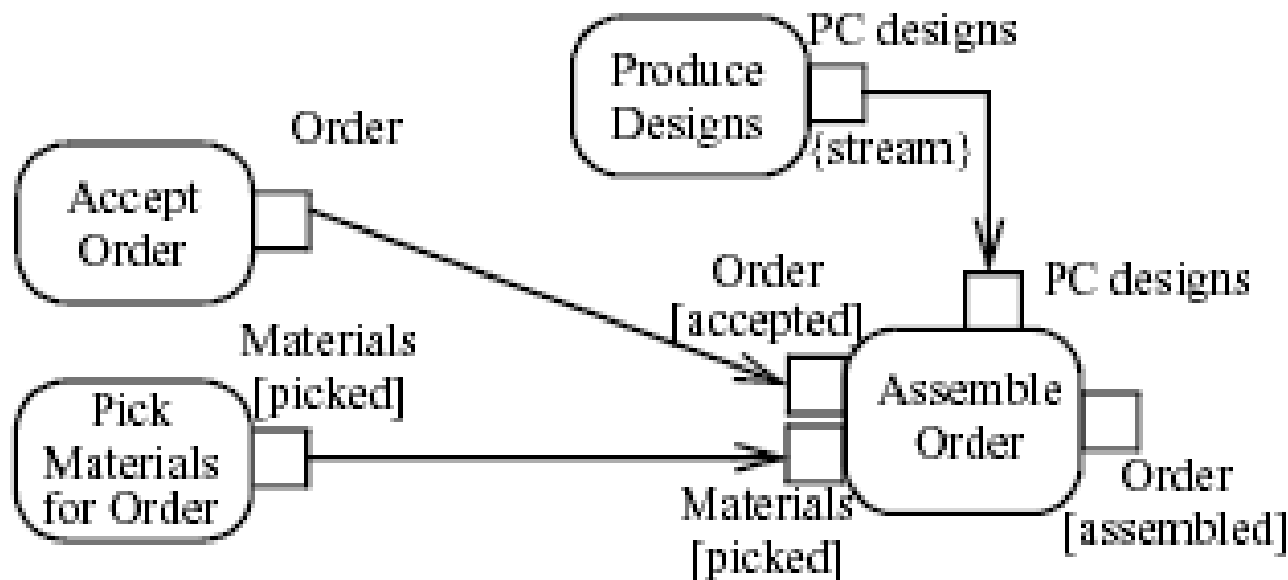
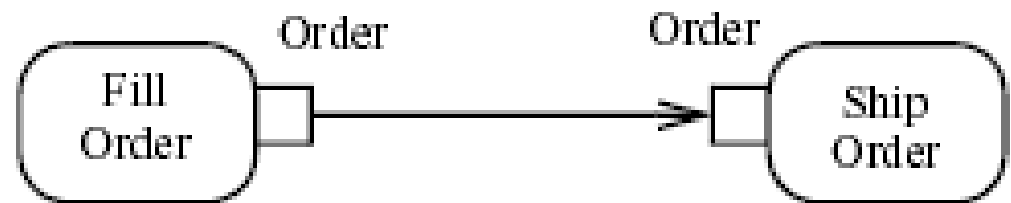
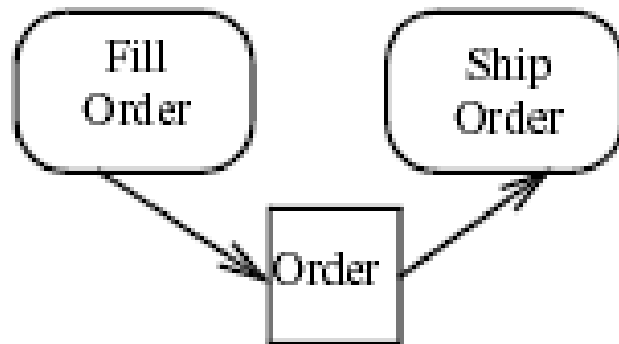


*Input pin,
pin-style, streaming*

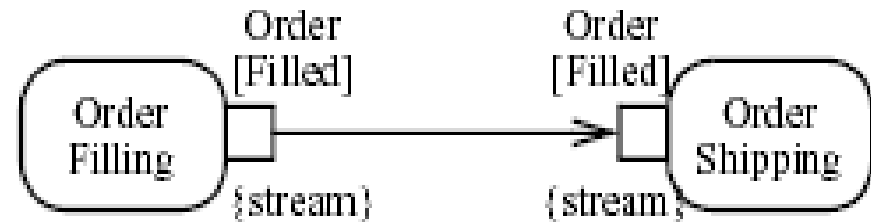
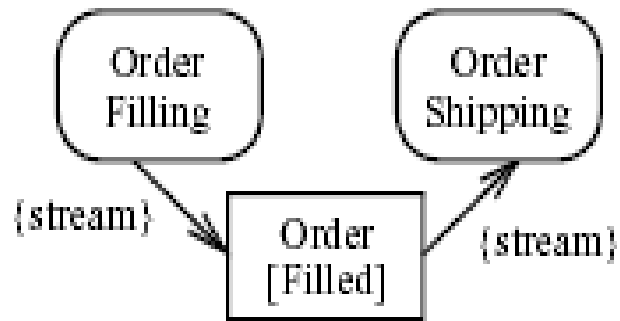


*Output pin,
pin-style, streaming*

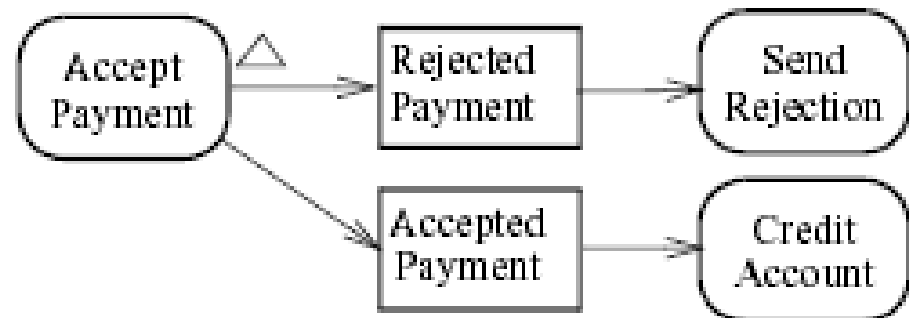
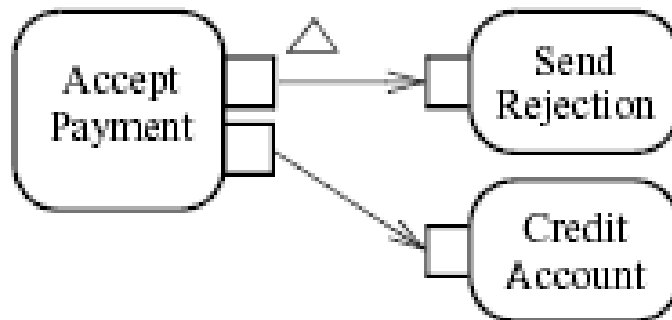
Ejemplos pines(I)



Ejemplos pines(II)

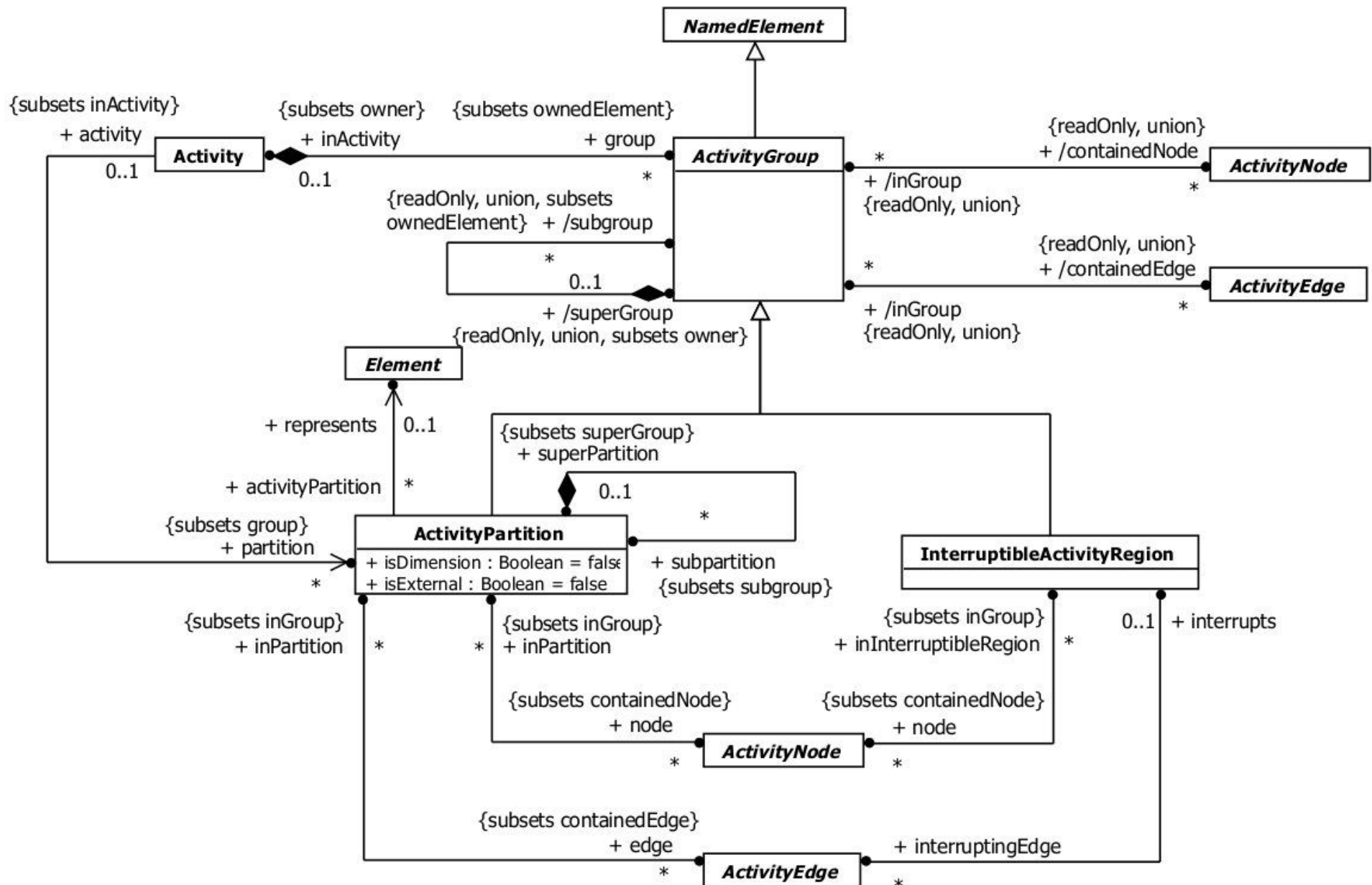


Flujos

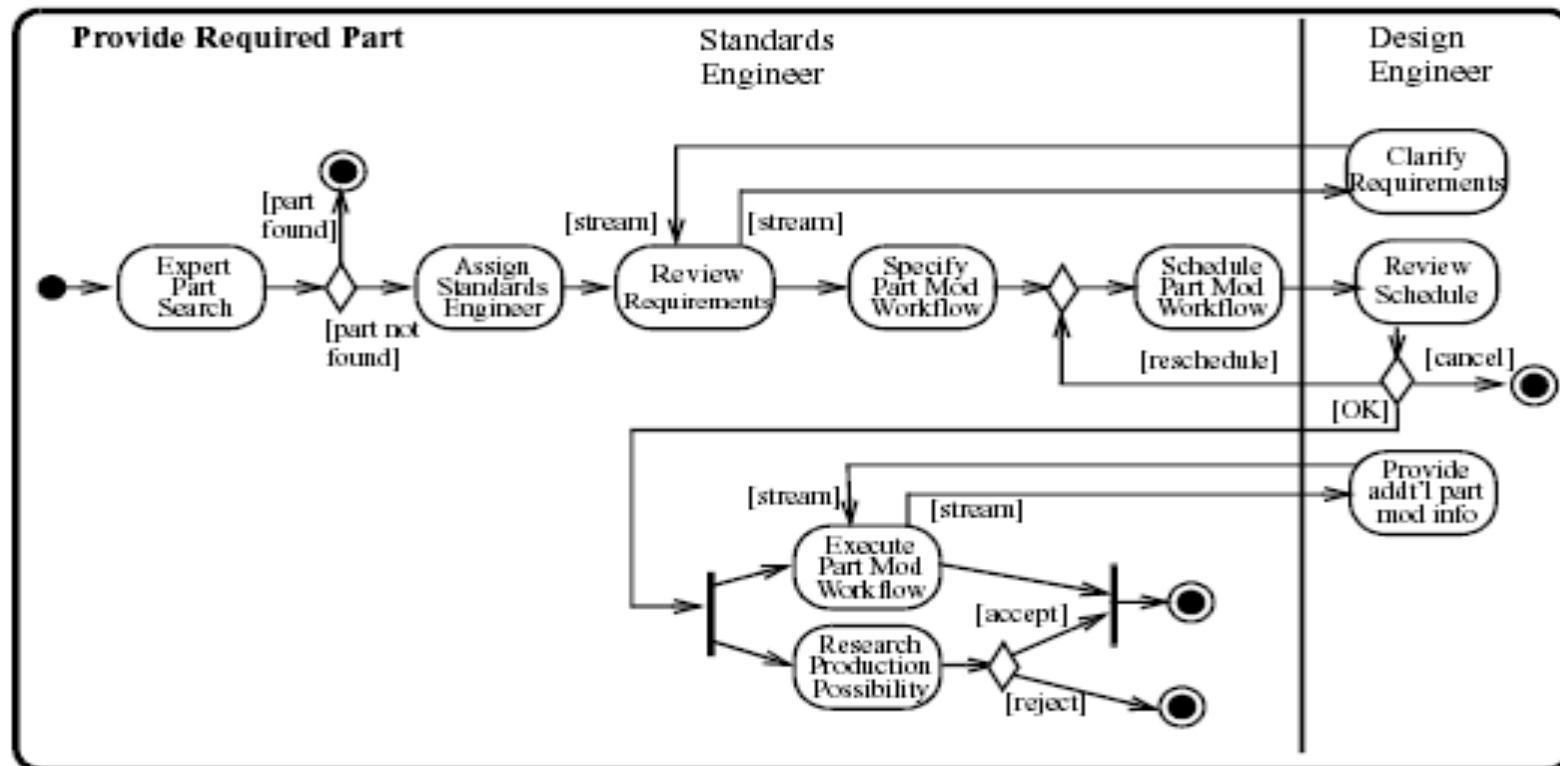
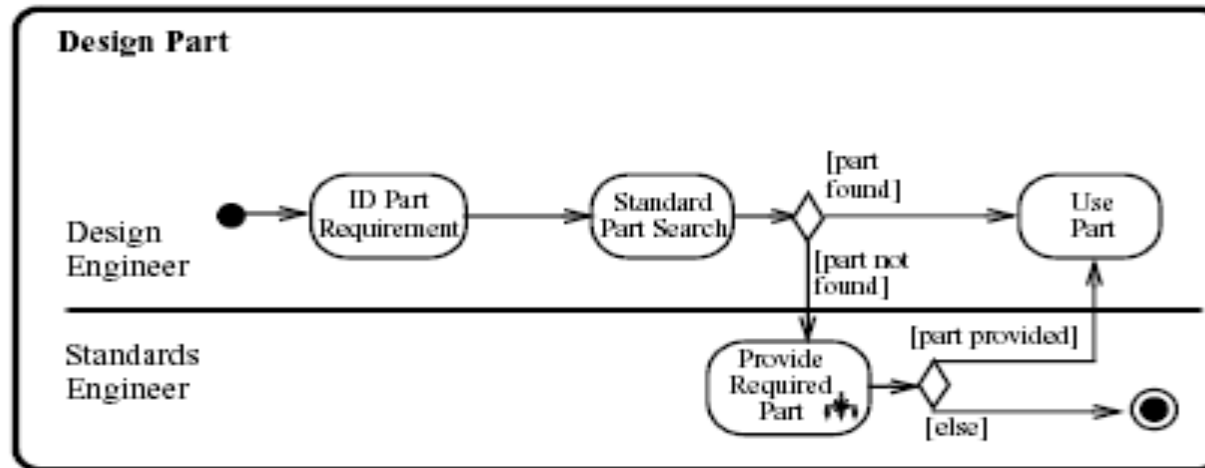


Excepciones

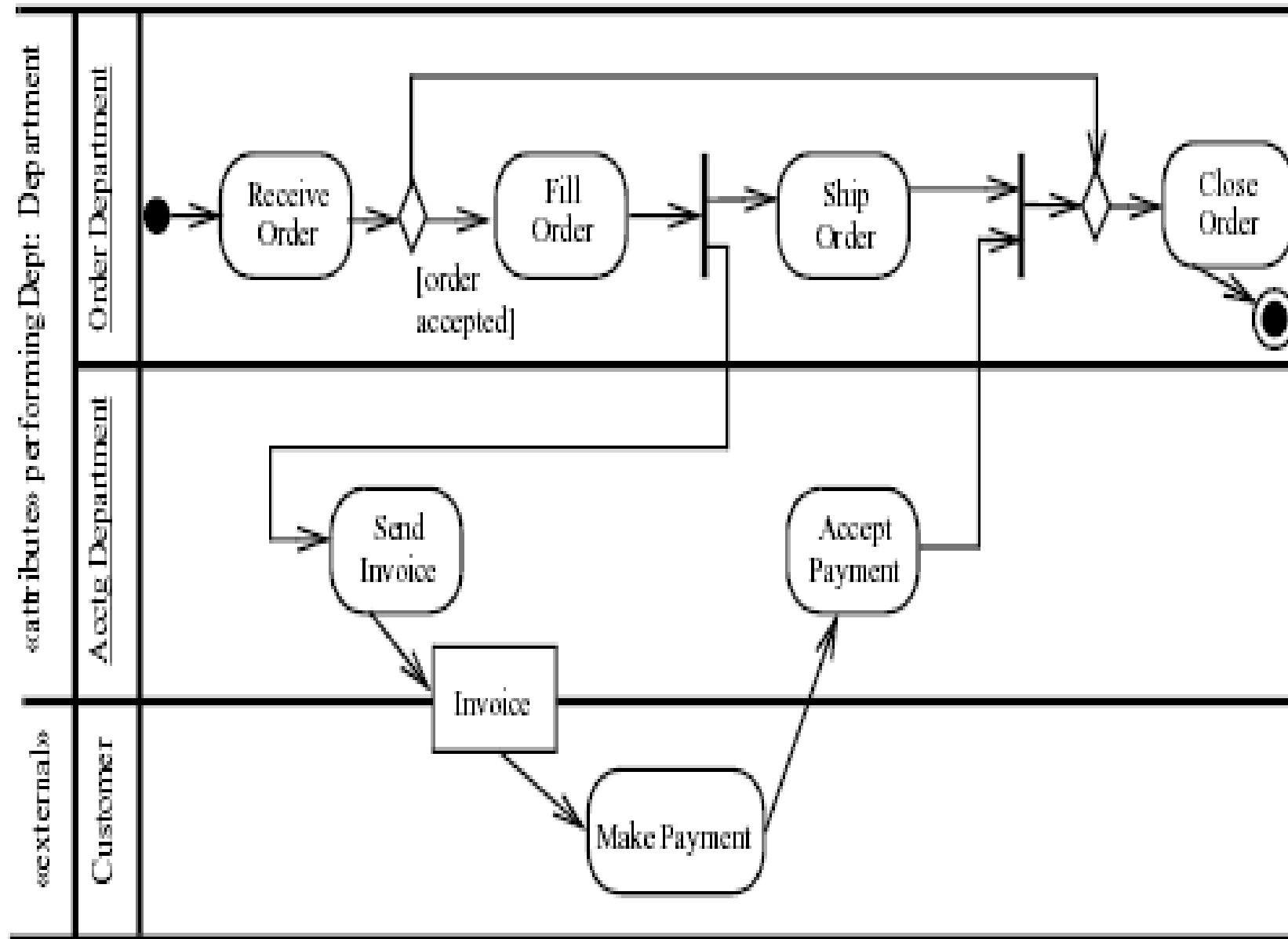
Sintaxis abstracta de grupos de actividades



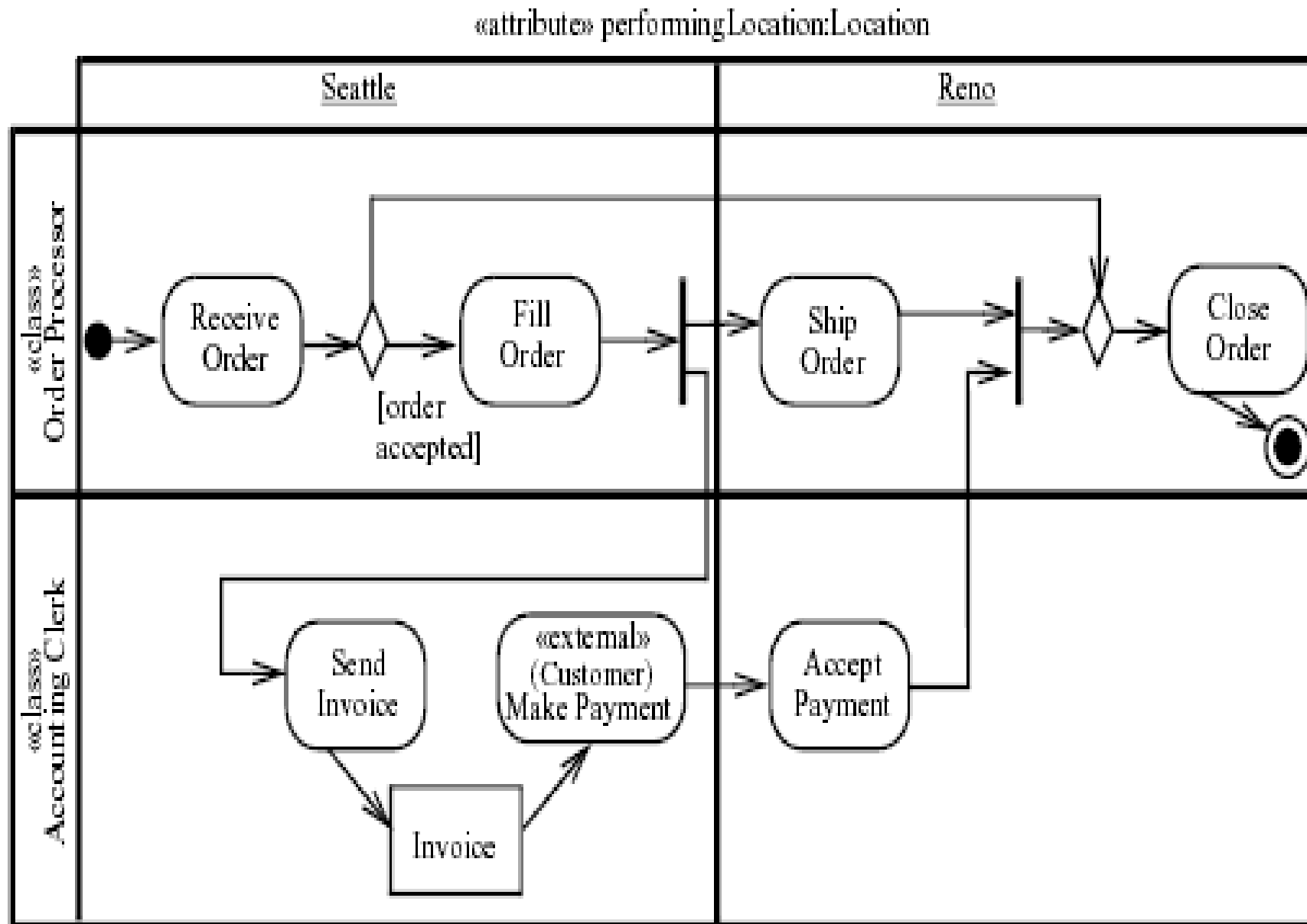
Flujo de trabajo(I)



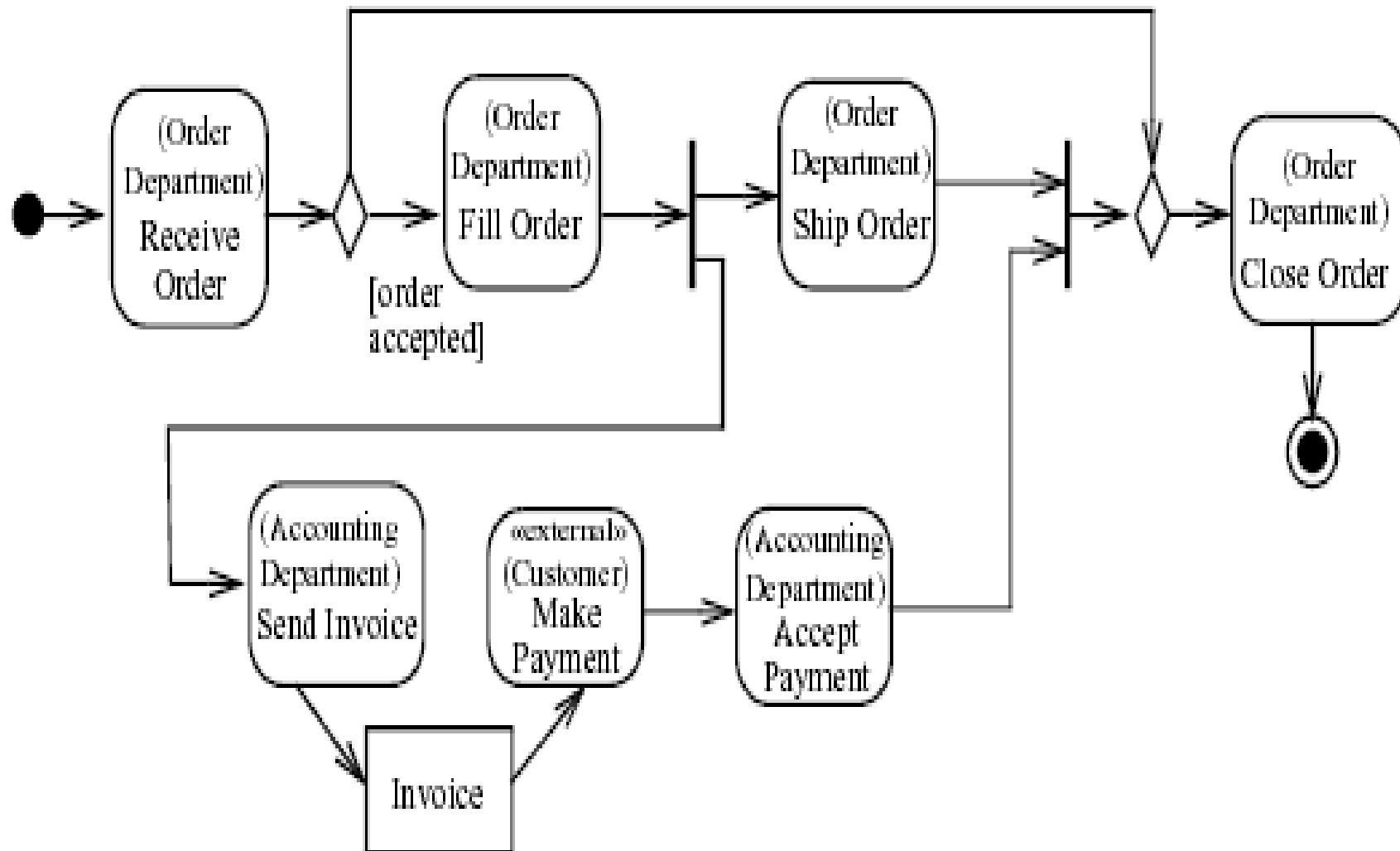
“Swimlane Partition” de actividades



“Multidimensional Swimlane Partition”



(“Annotated Partition”) de actividades (II)

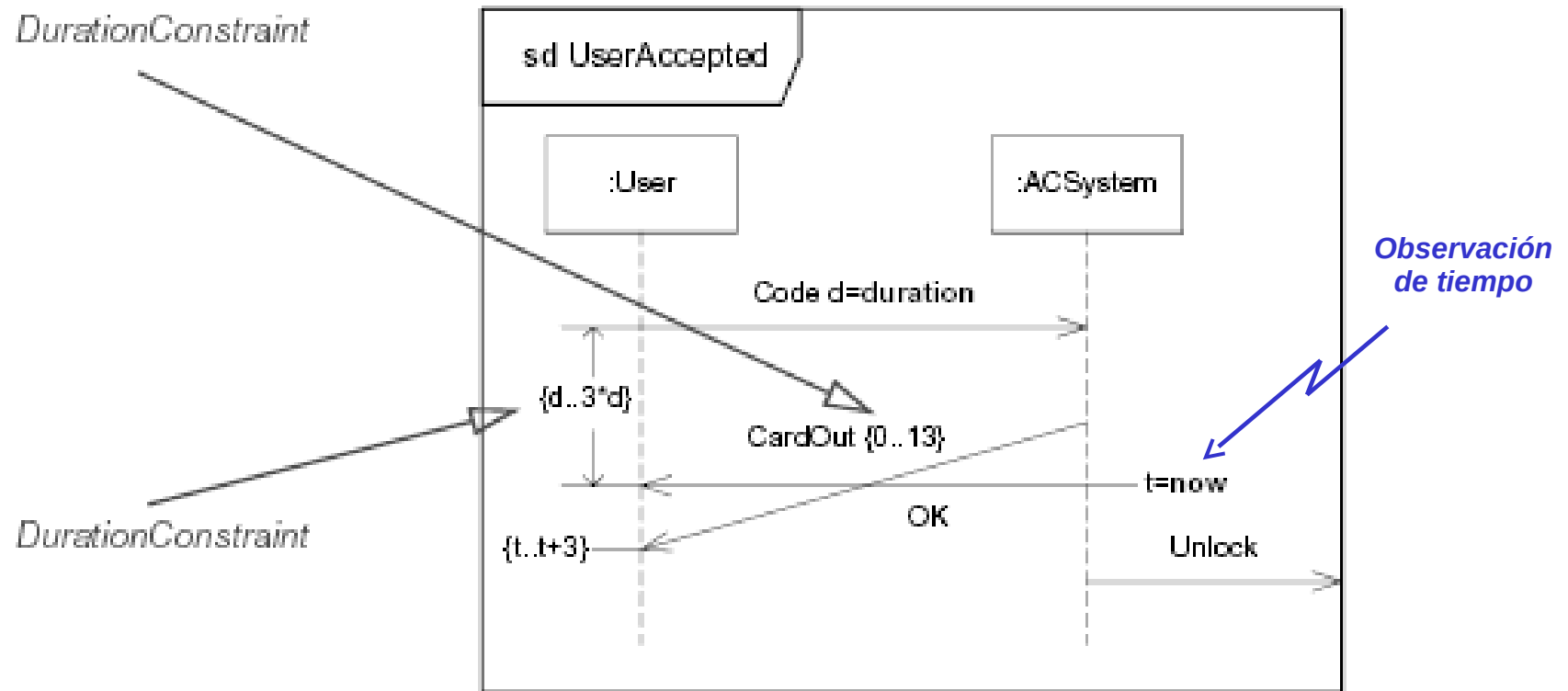


Algunos conceptos básicos de comportamiento

Clase(Objetos) activa(activos) y tiempo



Clase activa



Tiempo y restricciones de tiempo