



Ingeniería de Software

Ingeniería de Requisitos Parte 3

- **Un caso de uso describe una secuencia de interacciones entre un sistema y un actor externo que resultan en un resultado de valor para el actor.**
- Cada escenario comprende una instancia de uso del sistema. Un caso de uso contempla un conjunto de escenarios relacionados.
- Son independientes del método de diseño y del lenguaje que se utilice para la implementación.

Actores

- Un actor es una persona (o quizás otro sistema) que interactúa con el sistema para realizar un caso de uso.

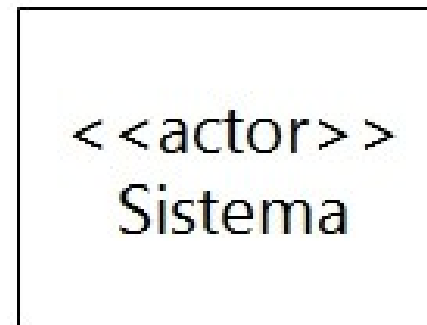
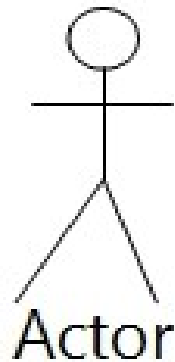
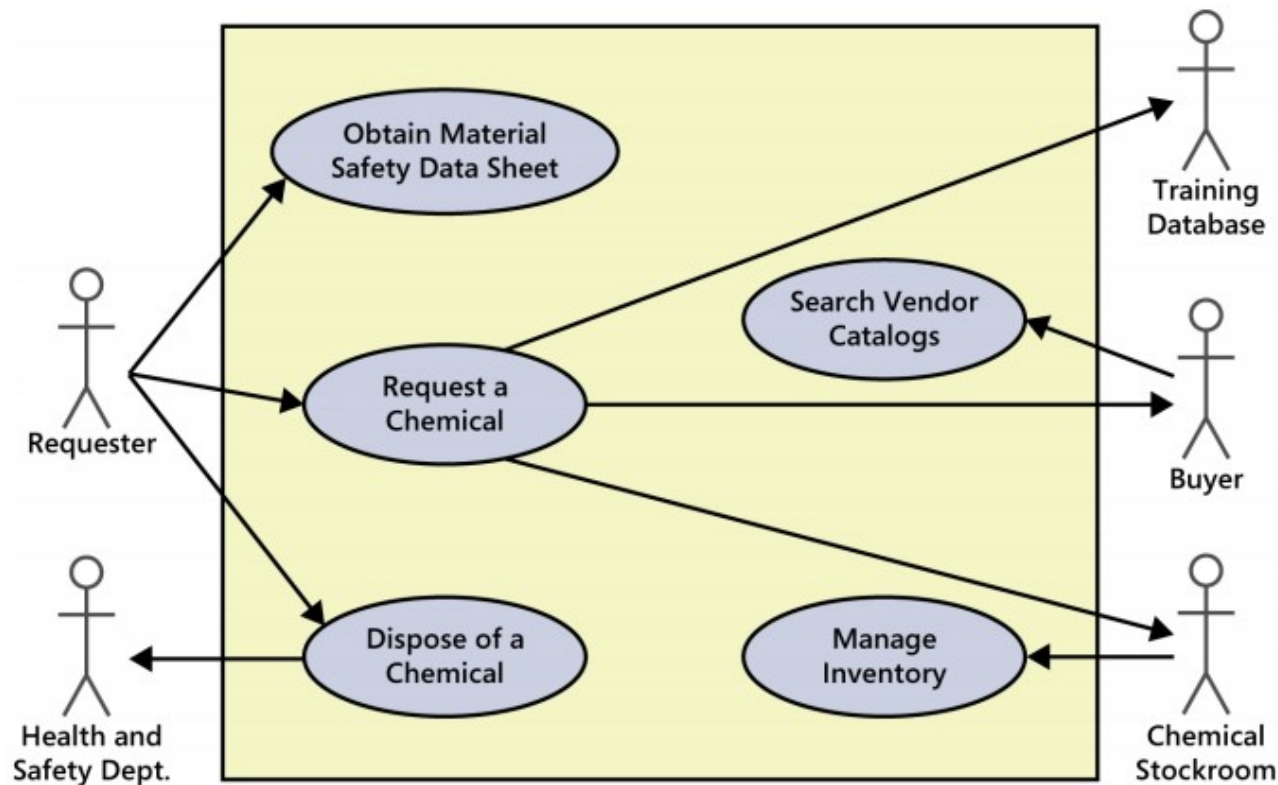


Diagrama de Casos de Uso

- Provee una representación de alto nivel de los requisitos de usuario.



Elementos esenciales de un Caso de Uso

- Identificador único.
- Breve descripción.
- Precondiciones.
- Poscondiciones.
- *Una lista enumerada de pasos que muestran la secuencia de interacciones entre el actor y el sistema.*

Descripción

- Es una descripción breve, a alto nivel, del caso de uso y del flujo de acciones que comprende.
- Ejemplo: *Realizar Transferencia* - Descripción
 - Este caso de uso comienza cuando un usuario se loguea al sistema y le indica que desea realizar una transferencia.
El usuario indica las cuentas de origen y destino de la transferencia y el monto de la misma. El sistema se comunica con el banco, el cual realiza la transferencia especificada.

Precondiciones

- Describen condiciones que debe cumplir el sistema para que se pueda iniciar el caso de uso.
- Ejemplo: *Impresión de Factura* - Precondiciones
 - El proceso de liquidación de la factura ha sido iniciado. La factura se encuentra pendiente de impresión.

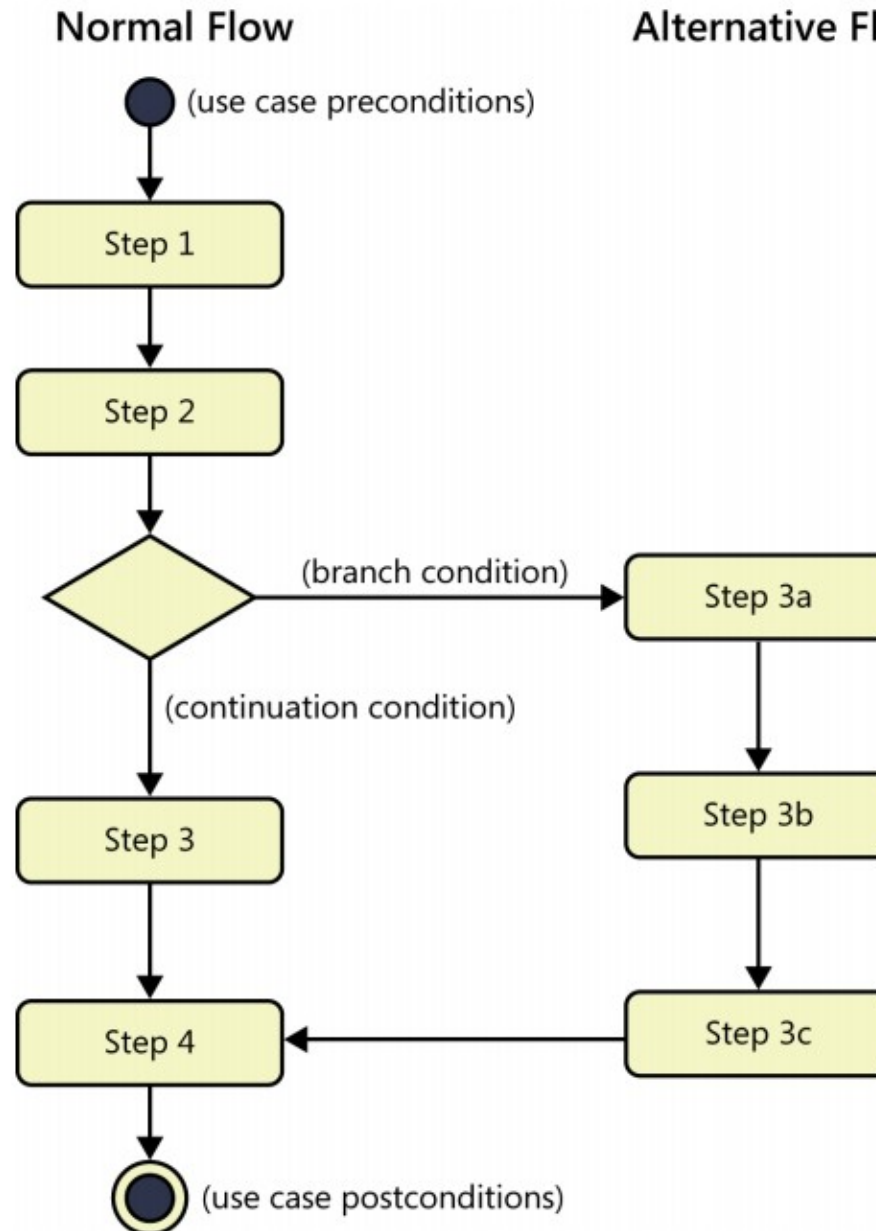
Poscondiciones

- Las poscondiciones describen el estado del sistema luego de la ejecución exitosa del caso de uso.
- Ejemplo: *Balance Contable Anual*- Poscondiciones
 - El ejercicio contable queda cerrado, esto implica que a partir de ese momento no es posible registrar nuevos asientos contables para ese ejercicio.

Flujo normal y flujos alternativos

- El flujo principal muestra la secuencia de pasos más común que se lleva a cabo para la ejecución del caso de uso.
- Además del flujo principal, un caso de uso puede tener varios flujos alternativos. Un flujo alternativo describe un escenario alternativo al principal (casos particulares y manejo de excepciones/errores).

Flujo normal y flujos alternativos



Ejemplo de flujo normal

1. El usuario ingresa al sitio.
2. El sistema solicita que se registre.
3. El usuario ingresa sus datos.
4. El sistema envía sus datos al banco.
5. El banco valida los datos.
6. El sistema le permite al usuario ingresar y le ofrece un menú de servicios.
7. Usuario: indica al sistema que desea realizar una transferencia
8. Sistema: lista las cuentas del usuario
9. Sistema: pide al usuario que seleccione la cuenta de origen de la transferencia de la lista o que la ingrese manualmente.
10. Usuario: selecciona cuenta de origen de la transferencia de la lista.
11. Sistema: pide al usuario que seleccione la cuenta de destino de la transferencia de la lista o que la ingrese manualmente.
12. Usuario: selecciona cuenta de destino de la transferencia de la lista.
13. Sistema: pide al usuario que ingrese el monto de la transferencia.
14. Usuario: ingresa el monto de la transferencia.
15. Sistema: se comunica con el servidor del banco y le indica que desea realizar una transferencia entre las cuentas especificadas y con el monto indicado.
16. Banco: realiza la transferencia especificada e indica que la transferencia se realizó con éxito.
17. Sistema: actualiza el monto de las cuentas involucradas.
18. Sistema: notifica al usuario que la transferencia se efectuó correctamente.

Ejemplos de flujos alternativos

10.A. El usuario desea ingresar manualmente el número de cuenta origen.

- 10.A.1. Usuario: selecciona ingresar manualmente el número de cuenta.
- 10.A.2. Sistema: pide al usuario que ingrese número de cuenta y código del banco.
- 10.A.3. Usuario: ingresa número de cuenta y código del banco.
- 10.A.4. Vuelve al punto 5.

16.A. Existe un error en la comunicación con el banco.

- 16.A.1. Sistema: no se puede comunicar con el banco
- 16.A.2. Sistema: informa al usuario que no se puede comunicar con el banco en este momento.
- 16.A.3. Fin CU

Flujos alternativos de flujos alternativos

12.B.4.A. Condición que lleva a este flujo alternativo

12.B.4.A.1. Paso 1

12.B.4.A.2. Paso 2

...

o bien

12.B.4.A. Condición que lleva a este flujo alternativo

1. Paso 1

2. Paso 2

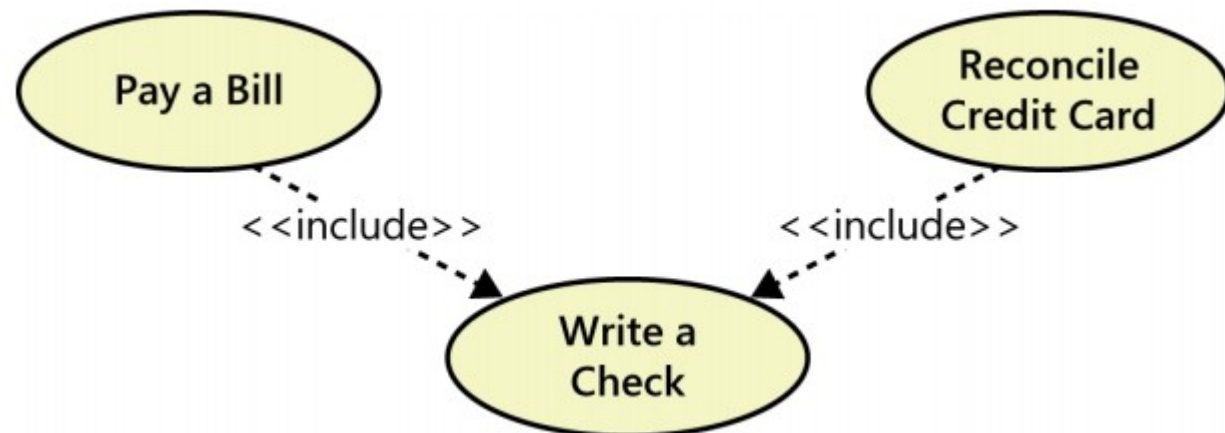
...

Casos de uso expandidos

- Usualmente se crea una primera versión del caso de uso que no contempla los flujos normales y alternativos, sino que incluye únicamente descripción, precondiciones y poscondiciones. A este estilo de escritura se le llama **alto nivel**.
- En estos casos a la versión que incluye los flujos normales y alternativos se le llama **caso de uso expandido**.

Relaciones entre CUs - Inclusión

- Permiten definir casos de uso que “ejecutan” otros casos de uso.
- Al caso de uso que incluye a otro se le llama caso base y al otro se le denomina caso incluido.
- Se puede incluir otros casos de uso tanto en el flujo principal como en algún flujo alternativo.



Relaciones entre CUs - Inclusión

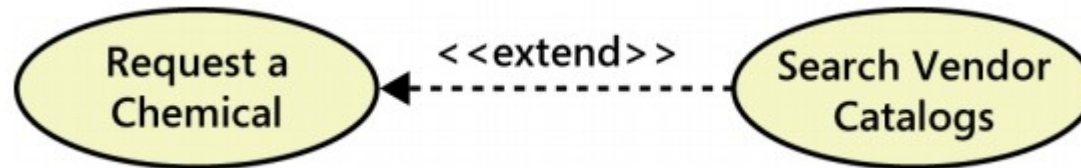
- Es como ejecutar la totalidad del caso de uso incluido y luego continuar en el punto siguiente del caso base.
- Sólo se continúa la ejecución del caso base si el caso incluido se ejecutó con éxito.

1. El usuario ingresa al sitio.
2. <<incluye>> "Autentificar"
3. El sistema le ofrece un menú de servicios.
4. Usuario: indica al sistema que desea realizar una transferencia
5. Sistema: lista las cuentas del usuario
6. ...

Relaciones entre CUs - Extensión

- La relación de extensión refiere a un fragmento de un caso de uso que extiende, agrega comportamiento, a otro caso de uso.
- Se utilizan para describir escenarios alternativos complejos.
- Una extensión sólo se ejecuta cuando se cumple una condición particular en un punto específico del caso de uso a extender. A ese punto se le llama punto de extensión.

Relaciones entre CUs - Extensión



- En el caso de ejemplo, *"Request a Chemical"* tiene como objetivo solicitar un químico de la proveeduría y tiene como alternativo solicitar el químico de un vendedor.
- El solicitante puede ejecutar el caso de uso *"Search Vendor Catalogs"* cuando solicita un químico, como parte del procesamiento del flujo alternativo.

Relaciones entre CUs – Extensión – Ejemplo

- **Caso de uso:** Retirar dinero de cajero
- **Flujo principal:**
 1. Incluye el caso de uso: Identificar Cliente
 2. El CA despliega las distintas alternativas disponibles: retiro, depósito, consulta
 3. El Cliente elige Retiro
 4. El CA pide cuenta y monto
 5. El Cliente los ingresa
 6. CA envía código de Tarjeta, PIN, cuenta y monto al SC
 7. El SC contesta: OK
 8. El Cliente pide dispensar el dinero
 9. El CA dispensa el dinero
 10. El CA devuelve la tarjeta
 11. El CA imprime el recibo
- **Puntos de Extensión:**
 - Retiro de Monedas: En el punto 8 del flujo principal

Relaciones entre CUs – Extensión – Ejemplo

- **Caso de Uso:** Retirar monedas
- **Descripción Breve:** El cliente opcionalmente puede querer retirar monedas.
- **Flujo Principal:**
Extensión de Retirar en el punto Retirar Monedas, el cliente también puede elegir “monedas”, en ese caso:
 1. El Cliente elige retirar monedas, especificando tipos de monedas y la cantidad de rollos para cada uno.
 2. El CA calcula el importe a retirar para cada moneda y el total y lo muestra.
 3. El Cliente confirma.

Estrategias para identificar casos de uso

- Identificar primero a los actores, luego a los procesos que serán soportados por el sistema y por último definir casos de uso para las actividades en las cuales interactúan los actores y el sistema.
- Crear escenarios específicos para ilustrar cada proceso de negocio, luego generalizar en casos de uso e identificar actores.
- Identificar los eventos externos a los cuales el sistema debe responder, luego relacionar esos eventos a los actores participantes y casos de uso específicos.
- Identificar que entidades de datos necesitan casos de uso de creación, lectura, actualización, borrado u otras operaciones de manipulación.

Consideraciones

- No tratar de forzar a que todos los requisitos sean contemplados por los casos de uso. En algunos casos no aporta demasiado valor hacerlo.
- Ser concretos y mantenerse dentro de los límites de los casos de uso (precondiciones y poscondiciones).
- Evitar escribir demasiados casos de uso. Evaluar si son todos necesarios y si no corresponden a distintos escenarios de un mismo caso de uso.
- No crear casos de uso muy complejos. En particular, no incluir demasiados chequeos de errores y validaciones comunes.

Consideraciones

- Los casos de uso no tienen porque desarrollarse por completo de una vez. En particular, no utilizar demasiado tiempo en detallar casos de uso que serán utilizados como insumo para otras actividades que comienzan dentro de meses o años.
- No se deberían incluir en los flujos detalles de la interfaz gráfica. Pero pueden agregarse secciones, por ejemplo, para incluir borradores de pantallas.
- No se debería incluir información de diseño o desarrollo. Así tampoco información sobre la definición de los datos.
- No escribir casos de uso que no sean entendidos por los usuarios.

Modelado del sistema

- El modelado del sistema es el proceso de desarrollar modelos abstractos del sistema, con cada modelo se presentan diferentes visiones o perspectivas.
- Ayuda a los analistas a entender las funcionalidades del sistema y facilita la comunicación con los clientes.
- En la actualidad es una actividad que se basa generalmente en la notación UML.

Modelado de sistemas existentes y nuevos

- Los modelos de sistemas existentes se utilizan durante la ingeniería de requisitos. Ayudan a comprender qué hacen y pueden ser usados para discusiones acerca de sus fortalezas y debilidades.
- Los modelos de nuevos sistemas son usados durante la ingeniería de requisitos para ayudar a explicar los requisitos propuestos a otros stakeholders del sistema. También para discutir propuestas de diseño y documentar el sistema para su implementación.
- En un proceso de ingeniería guiado por modelos (model-driven), es posible generar de forma total o parcial una implementación del modelo del sistema.

Perspectivas del sistema - Ejemplo

- **Una perspectiva externa** — se modela el contexto o entorno del sistema.
- **Una perspectiva de la interacción** — se modelan las interacciones entre un sistema y su entorno o entre un sistema y sus componentes.
- **Una perspectiva estructural** — se modela la organización del sistema o la estructura de los datos que se procesan por el sistema.
- **Una perspectiva del comportamiento** — se modela el comportamiento dinámico del sistema y la forma en que responde a los eventos.

Perspectivas del sistema - UML

Tipos de diagramas considerados como esenciales (encuesta 2007)

- **Diagramas de actividad** — muestran las actividades involucradas en un proceso o en el procesamiento de datos.
- **Diagramas de casos de uso** — muestran las interacciones entre el sistema y su entorno.
- **Diagramas de secuencia** — muestran las interacciones entre el sistema y los componentes del sistema.
- **Diagrama de clases** — muestran las clases de objetos del sistema y las asociaciones entre ellas.
- **Diagramas de estados** — muestran como el sistema reacciona ante eventos internos o externos.

Uso de modelos gráficos

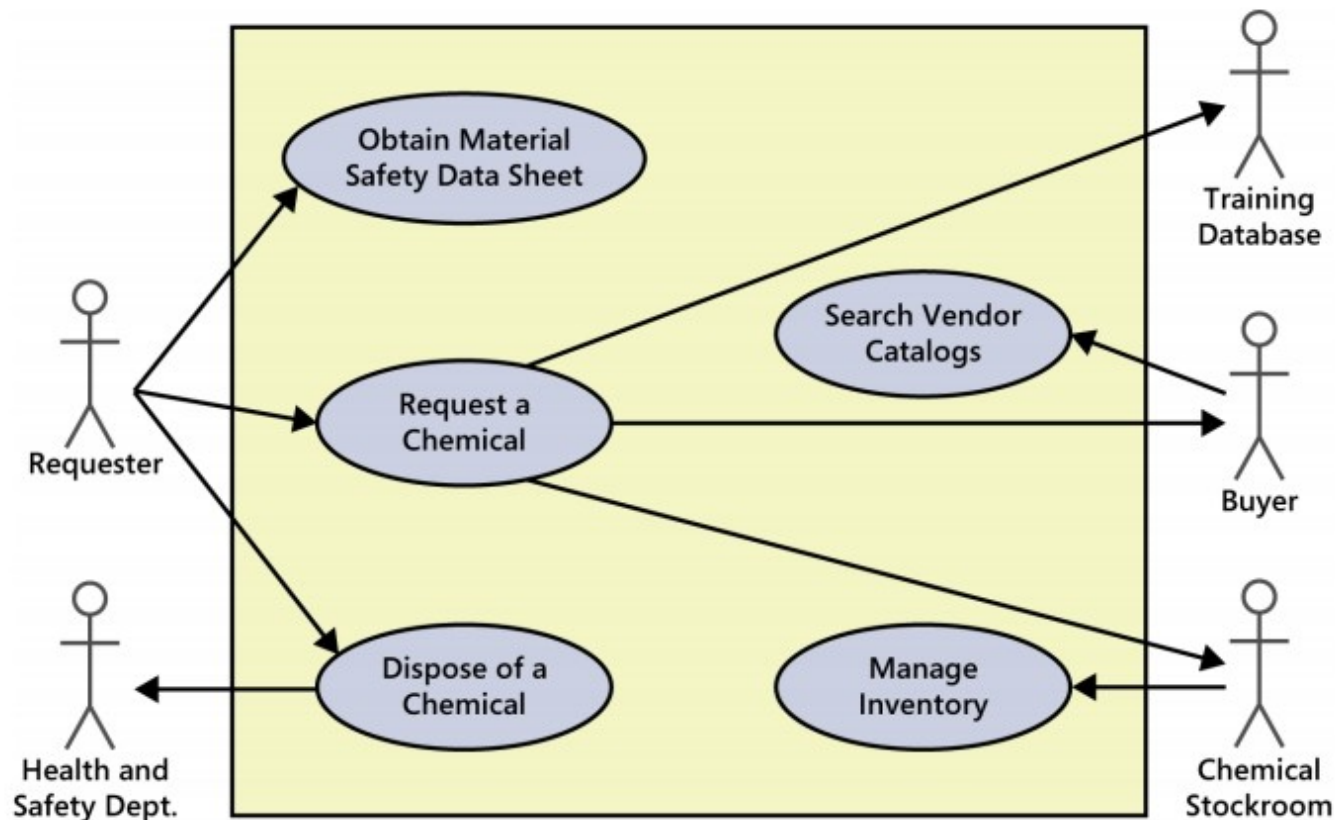
- Como medio de facilitar la discusión acerca de sistemas existentes o propuestos.
 - Ya sea que los modelos estén incompletos o incorrectos está bien debido a que su papel es aportar a la discusión.
- Como una manera de documentar un sistema existente.
 - Los modelos deben ser una representación exacta del sistema pero no tiene que ser completa.
- Como una descripción detallada del sistema que se puede usar para generar una implementación del sistema.
 - Los modelos tienen que ser a la vez correctos y completos.

Modelos de contexto

- Son usados para ilustrar el contexto operacional del sistema – muestran lo que está fuera de los límites del sistema.
- Temas sociales y organizacionales pueden afectar la decisión sobre donde situar los límites del sistema.
- **Límites del Sistema**
 - Definen qué está dentro y qué está fuera del sistema.
 - La definición de los límites del sistema tiene un profundo efecto sobre los requisitos del sistema.

Modelos de contexto – Ejemplo

- Podemos utilizar un diagrama de casos de uso que incluya la frontera del sistema como modelo de contexto.



Modelos de interacción

- Todo sistema involucra interacción: interacción con usuarios, con otros sistemas o entre sus componentes.
- Modelar la interacción con los usuarios ayuda a identificar los requisitos de usuario.
- Modelar la interacción con otros sistemas permite explorar posibles problemas de comunicación.
- Modelar la interacción entre los componentes ayuda a entender si la estructura propuesta del sistema es apropiada para los requisitos y atributos de calidad establecidos.

Modelos de interacción

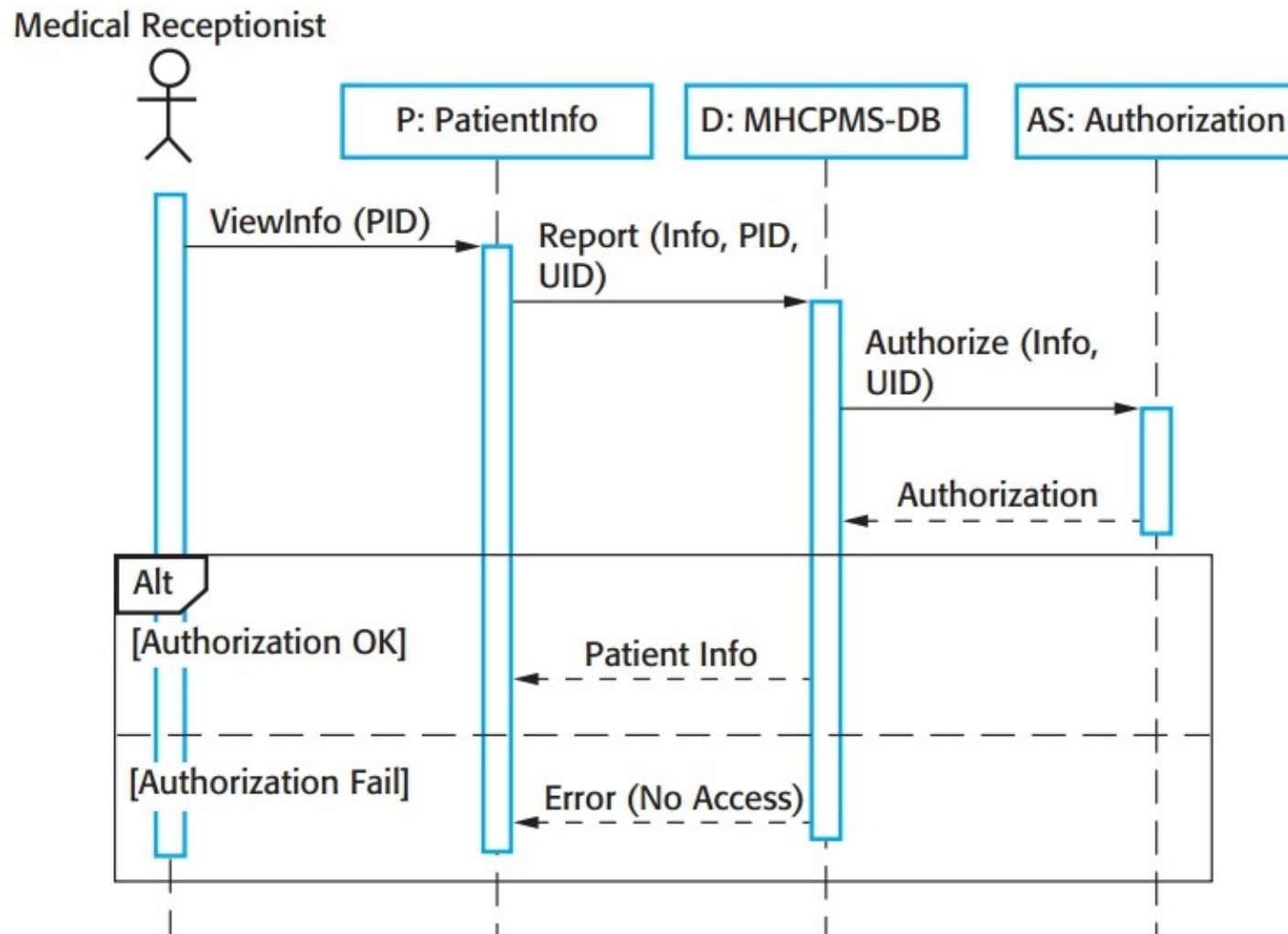
- En UML podemos modelar la interacción de un sistema utilizando:
 - Modelado con casos de uso.
 - Diagramas de secuencia.

Modelos de interacción

- **Diagramas de secuencia**

- Son usados para modelar las interacciones entre los actores y los objetos dentro de un sistema.
- Un diagrama de secuencia muestra la secuencia de interacciones que tienen lugar durante un caso de uso particular o una instancia de caso de uso.
- Los objetos y actores involucrados son enumerados a lo largo de la parte superior del diagrama con una línea vertical de puntos a partir desde allí hacía abajo.
- Las interacciones entre objetos son indicadas mediante flechas con anotaciones.

Modelos de interacción



Modelos estructurales

- Disponen la organización del sistema en términos de los componentes que lo componen y sus relaciones.
- Pueden ser modelos estáticos, los cuales muestran la estructura del sistema diseñado o modelos dinámicos los cuales muestran la organización del sistema cuando se está ejecutando.
- Se pueden crear modelos estructurales del sistema cuando se está discutiendo y diseñando la arquitectura del sistema.

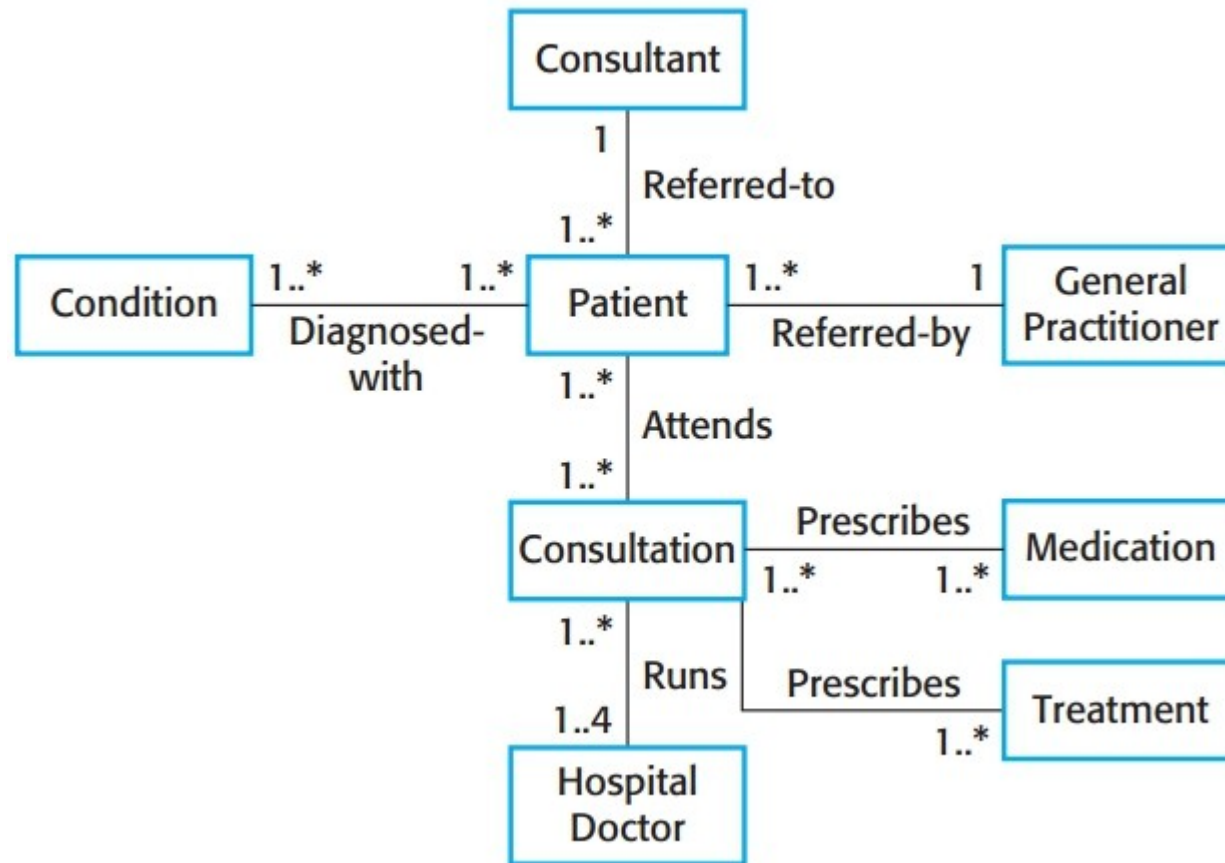
Modelos estructurales

- **Diagramas de clases**

- Muestran las clases de un sistema y sus asociaciones.
- Una clase puede ser considerada como una definición general de un tipo de objeto del sistema.
- Una asociación se representa como un enlace entre dos o más clases.
- Cuando se desarrollan modelos durante fases tempranas los objetos representan algo del mundo real (modelo conceptual).

Modelos estructurales

- Diagramas de clases – Ejemplo



Modelos de comportamiento

- Modelan el comportamiento dinámico de un sistema en ejecución.
 - Muestran qué sucede o qué debe suceder cuando un sistema responde a los estímulos de su entorno.
- Se puede pensar en dos tipos de estímulos:
 - **Datos** — Ciertos datos que llegan y deben ser procesados.
 - **Eventos** — Ciertos eventos que funcionan como disparadores.

Modelos de comportamiento

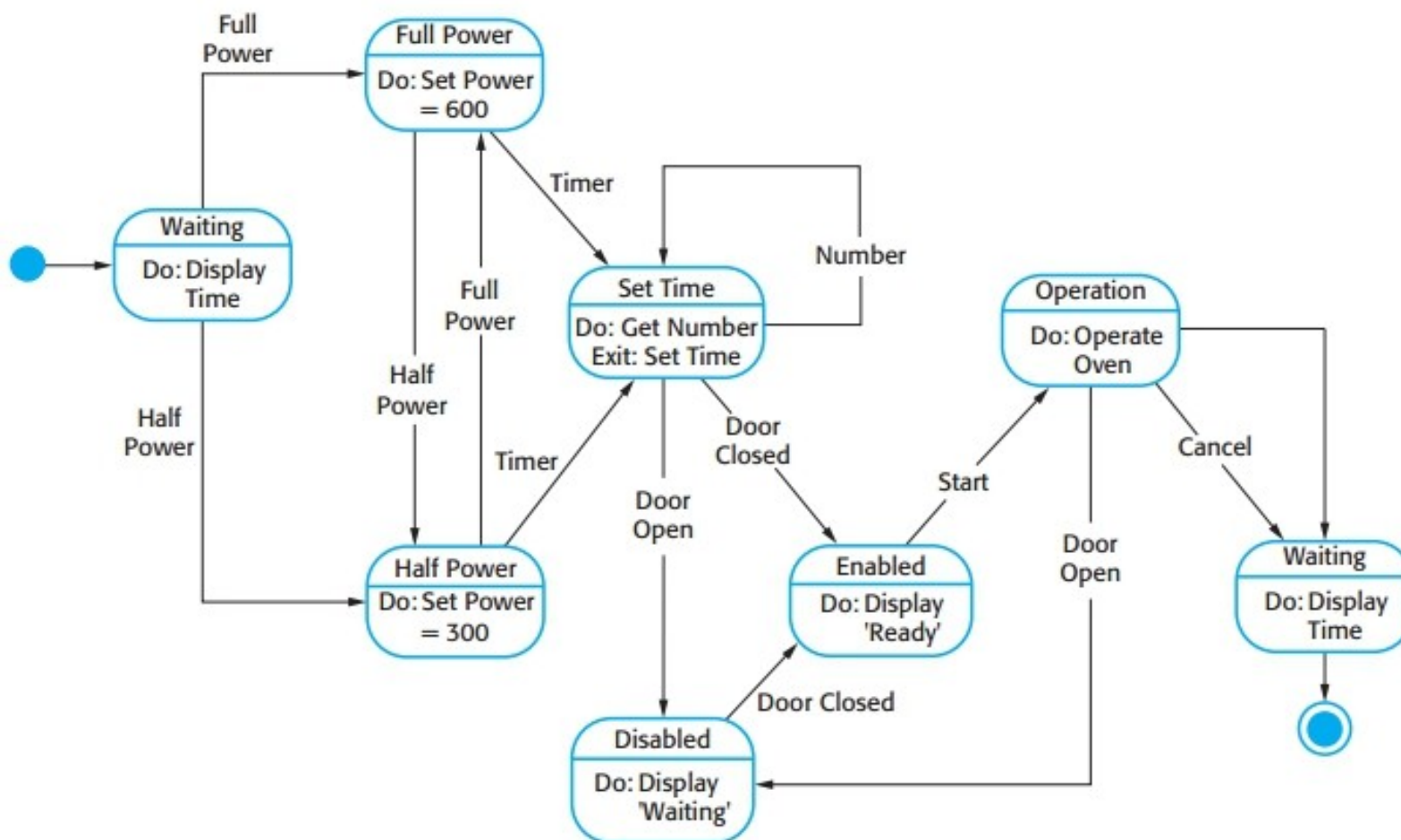
- **Modelado dirigido por datos (data-driven)**
 - Muestran la secuencia de acciones involucradas en procesar los datos de entrada y generar una salida asociada.
 - Los modelos de flujo de datos permiten realizar el seguimiento y documentar cómo los datos asociados a un proceso particular se mueven a través del sistema.
 - UML no soporta de forma natural los modelos de flujo de datos pero pueden utilizarse diagramas de secuencia.

Modelos de comportamiento

- **Modelado dirigido por eventos (event-driven)**
 - Muestran cómo un sistema responde a eventos externos e internos.
 - Se basan en la suposición de que un sistema tiene un número finito de estados y que eventos (estímulos) pueden causar una transacción de un estado a otro.
 - En UML se pueden utilizar diagramas de estado.

Modelos de comportamiento

- Diagrama de estado - ejemplo



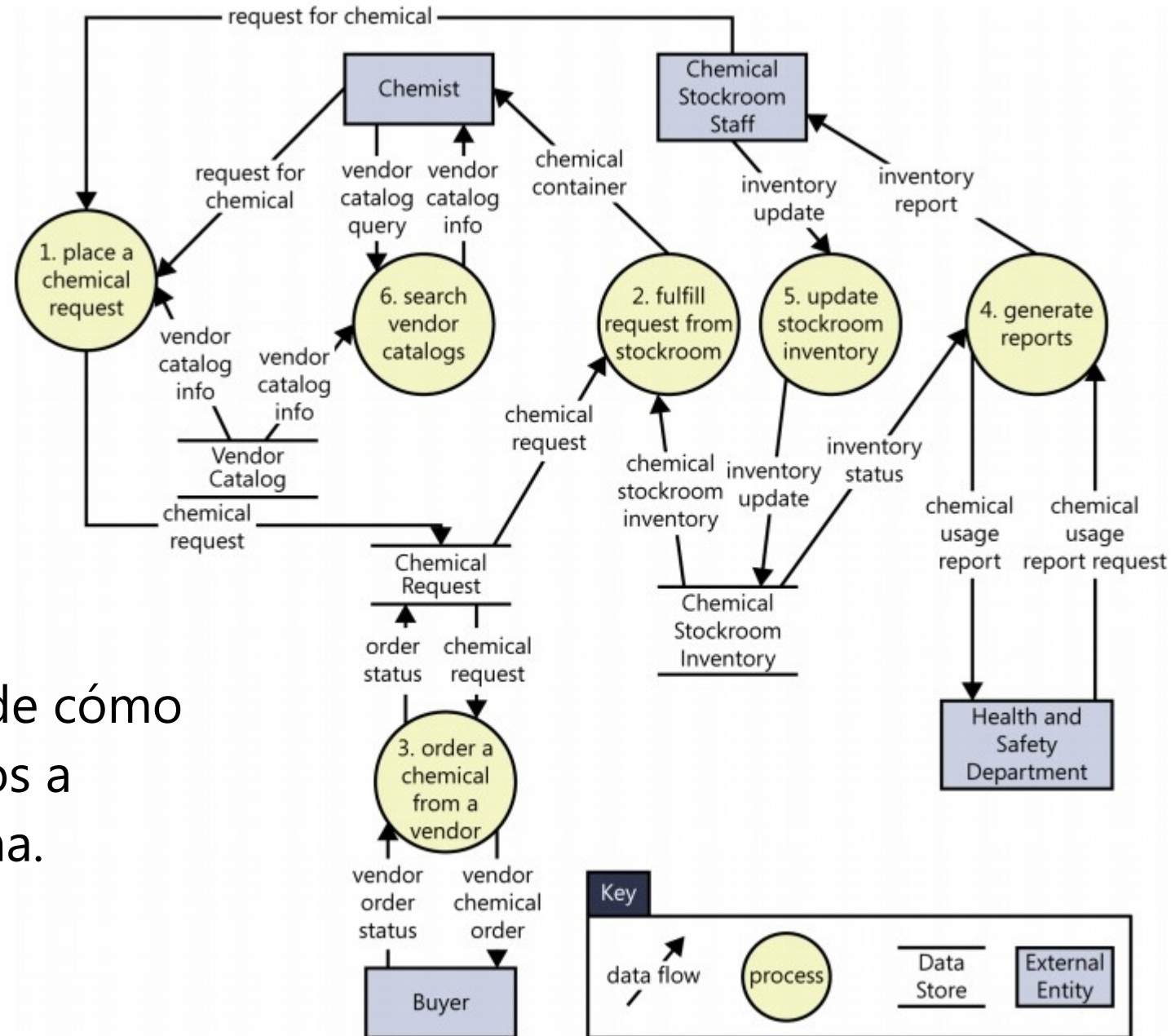
Ingeniería dirigida por modelos (MDE)

- Es un enfoque para el desarrollo de software donde los modelos (y no los programas) son los principales resultados del proceso de desarrollo.
- Los programas que se ejecutan en una plataforma de hardware/software se generan automáticamente a partir de los modelos.
- La MDE se encuentra todavía en una etapa temprana de desarrollo y no está claro si tendrá o no un efecto significativo en la práctica de la ingeniería de software.

Otros modelos para requisitos - Wieggers

- Los modelos visuales de requisitos ayudan a identificar requisitos faltantes, extraños o inconsistentes.
- Los modelos sirven tanto para elaborar y explorar requisitos así como para diseñar soluciones de software.
- No hay que asumir que los clientes saben cómo interpretar los modelos de análisis.
- En general no es posible modelar todo el sistema, así que lo recomendable es enfocarse en las porciones del sistema más riesgosas.

Diagrama de flujo de datos



- Provee una visión de cómo se mueven los datos a través de un sistema.

Diagrama de andariveles

- Proveen una manera de representar los pasos involucrados en un proceso o las operaciones de un nuevo sistema de software.
- Los carriles representan actores u otros sistemas que ejecutan un paso del proceso.
- Es uno de los modelos más simples de entender por los clientes.

Diagrama de andariveles

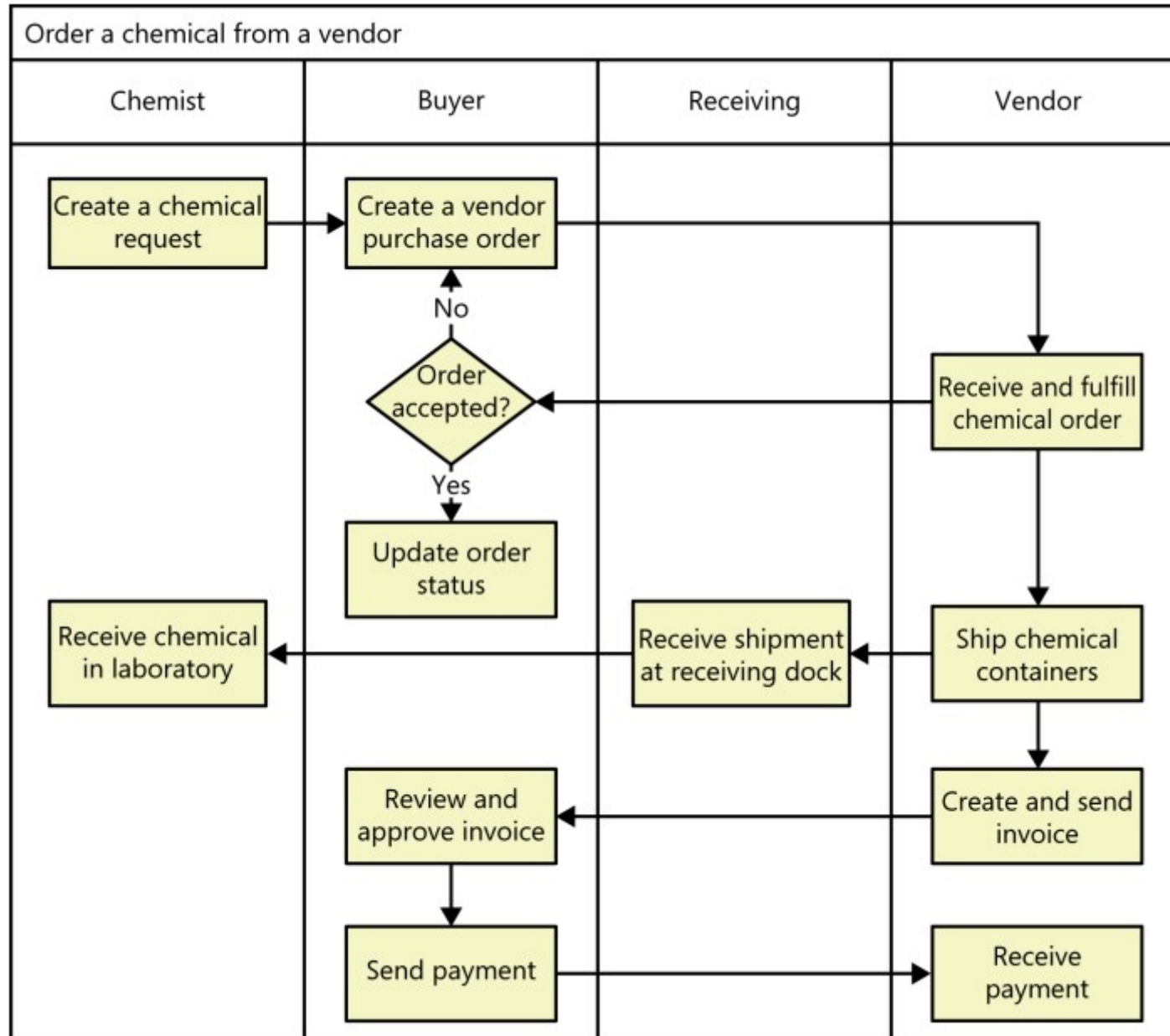


Diagrama de estados y transiciones y Tablas de estados

- Proveen una representación concisa, completa y no ambigua de los estados de un objeto o sistema.
- Los diagramas de estado contienen:
 - Los posibles estados se representan como rectángulos.
 - Las transiciones o los cambios de estado permitidos se representan como flechas entre dos rectángulos.
 - Los eventos o condiciones que causan los cambios de estado se muestran como etiquetas de texto sobre las flechas de transición.

Diagrama de estados y transiciones y Tablas de estados

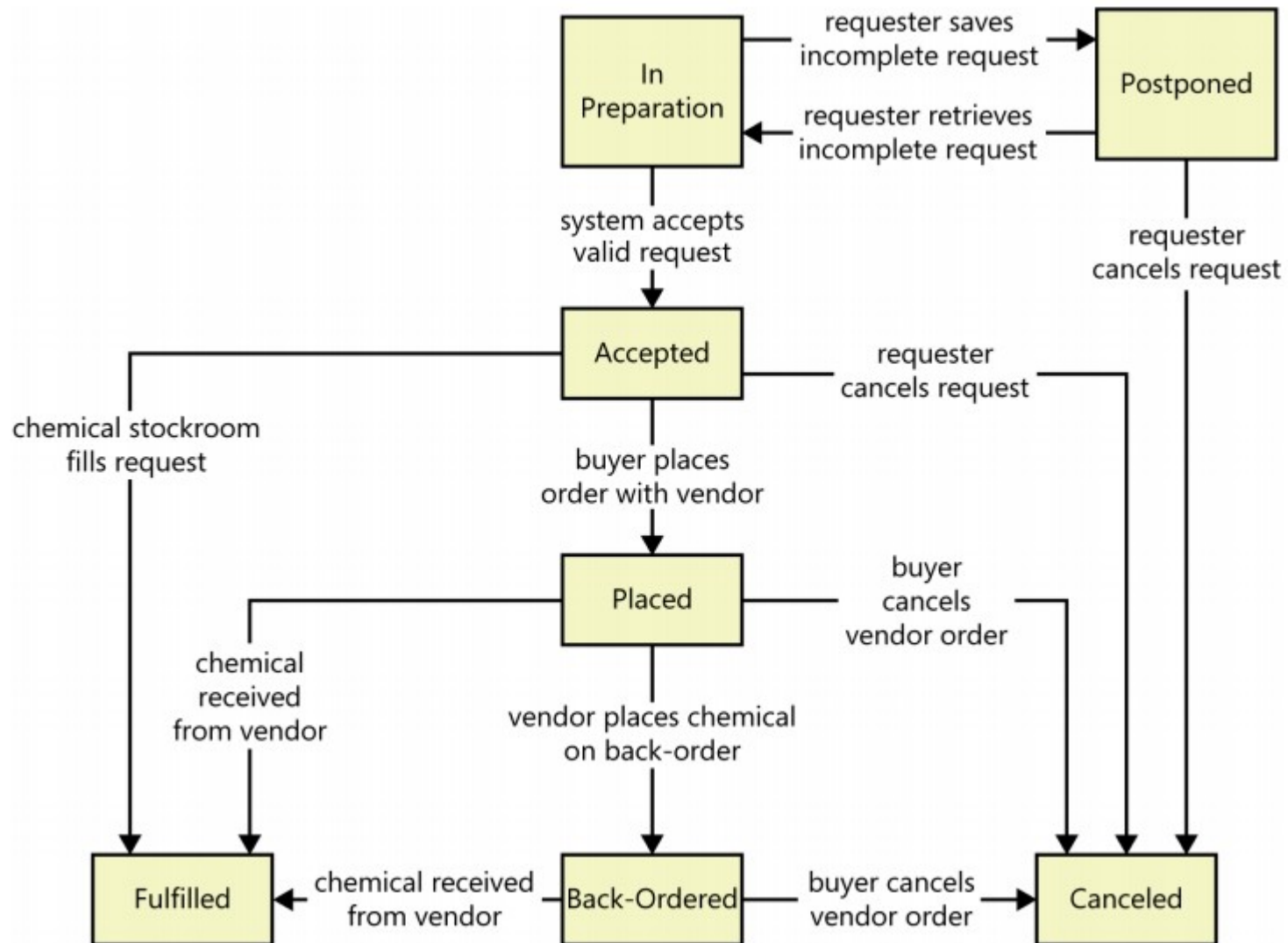


Diagrama de estados y transiciones y Tablas de estados

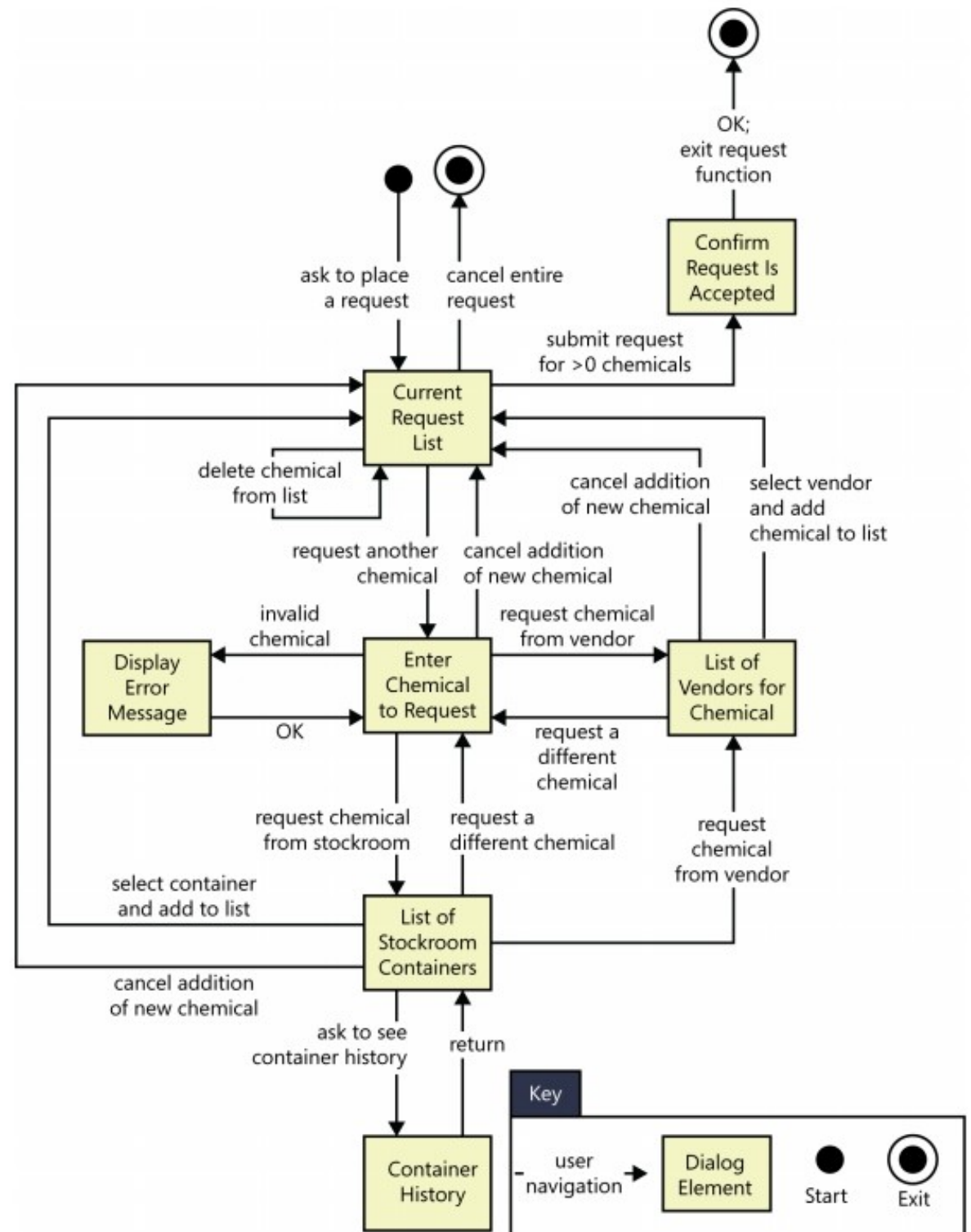
- Las tablas de estados muestran todas las posibles transiciones entre los estados en forma de una matriz.

	In preparation	Postponed	Accepted	Placed	Back-Ordered	Fulfilled	Canceled
In Preparation	no	user saves incomplete request	system accepts valid request	no	no	no	no
Postponed	user retrieves incomplete request	no	no	no	no	no	no
Accepted	no	no	no	buyer places order with vendor	no	chemical stockroom fills request	requester cancels request
Placed	no	no	no	no	vendor places chemical on back-order	chemical received from vendor	buyer cancels vendor order
Back-Ordered	no	no	no	no	no	chemical received from vendor	buyer cancels vendor order
Fulfilled	no	no	no	no	no	no	no
Canceled	no	no	no	no	no	no	no

Mapa de diálogo

- Un mapa de diálogo representa el diseño de la interfaz de usuario con un nivel alto de abstracción.
- La técnica permite modelar la interfaz de usuario en la forma de un diagrama de estados y transiciones.

Mapa de diálogo

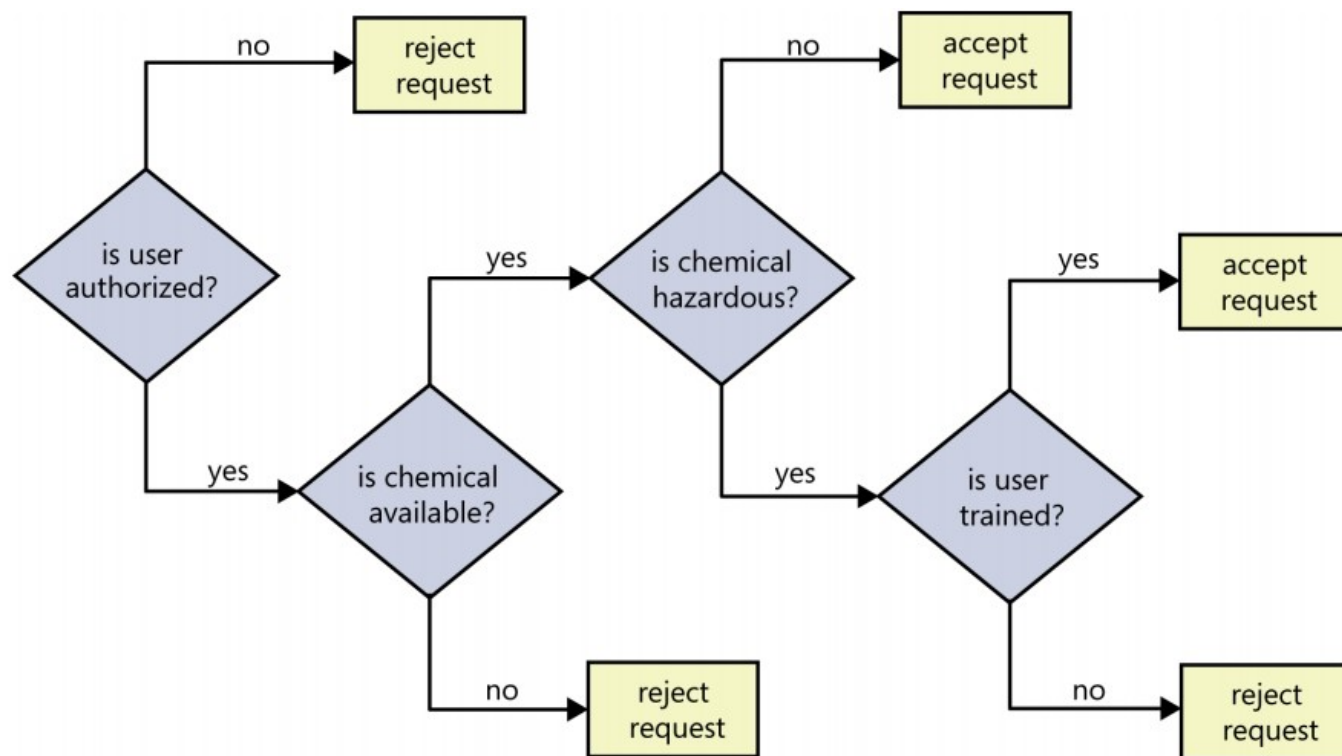


Tablas y árboles de decisión

- Las tablas y árboles de decisión son dos técnicas alternativas para representar qué debería hacer el sistema ante decisiones o lógica compleja.
- Una tabla de decisión lista los valores para todos los factores que influyen el comportamiento del sistema e indican la acción esperada del sistema en respuesta a cada combinación de factores.

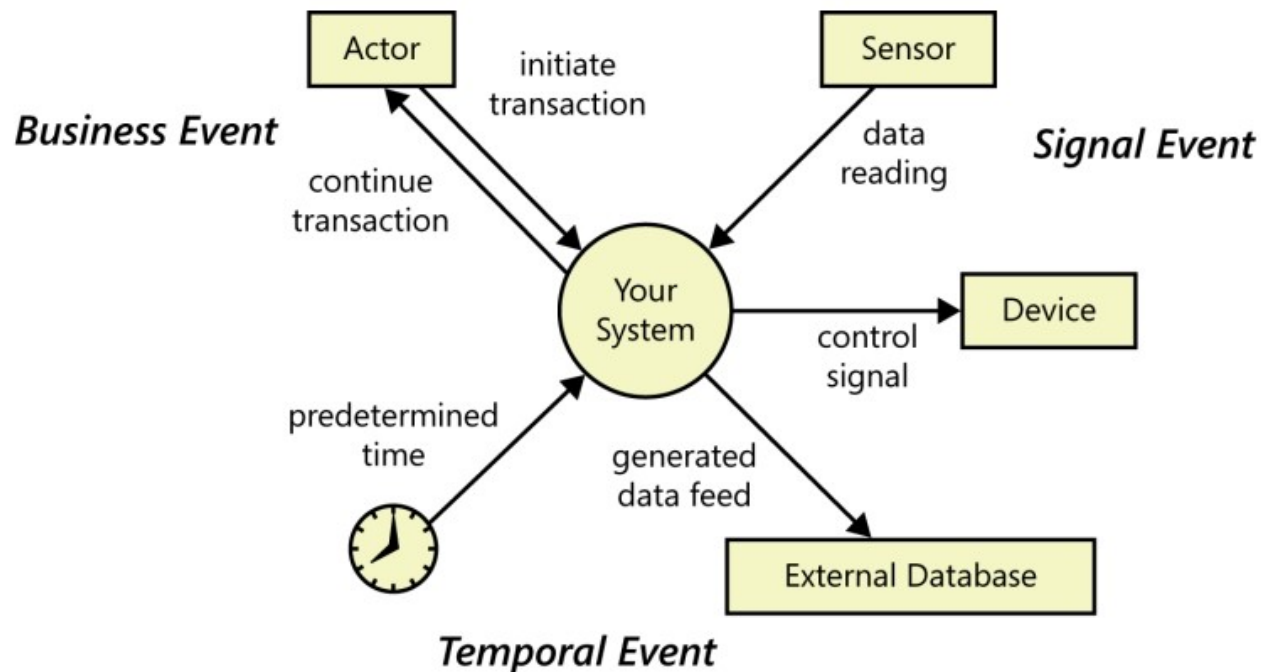
Tablas y árboles de decisión

Requirement Number					
Condition	1	2	3	4	5
User is authorized	F	T	T	T	T
Chemical is available	—	F	T	T	T
Chemical is hazardous	—	—	F	T	T
Requester is trained	—	—	—	F	T
Action					
Accept request			X		X
Reject request	X	X		X	



Tablas de evento-respuesta

- Permiten representar las respuestas del sistema a los posibles eventos que puedan ocurrir.
- Un evento es un cambio o actividad que tiene lugar en el ambiente del usuario y que estimula una respuesta del sistema.



Tablas de evento-respuesta

ID	Event	System state	System response
1	Set wiper control to low speed	Wiper off, on high speed, or on intermittent	Set wiper motor to low speed
2	Set wiper control to high speed	Wiper off, on low speed, or on intermittent	Set wiper motor to high speed
3	Set wiper control to off	Wiper on high speed, low speed, or intermittent	1. Complete current wipe cycle 2. Turn wiper motor off
4	Set wiper control to intermittent	Wiper off	1. Perform one wipe cycle 2. Read wipe time interval setting 3. Initialize wipe timer
5	Set wiper control to intermittent	Wiper on low speed or on high speed	1. Complete current wipe cycle 2. Read wipe time interval setting 3. Initialize wipe timer
6	Wipe time interval has passed since completing last cycle	Wiper on intermittent	Perform one wipe cycle at low speed setting
7	Change intermittent wiper interval	Wiper on intermittent	1. Read wipe time interval setting 2. Initialize wipe timer
8	Change intermittent wiper interval	Wiper off, on high speed, or on low speed	No response
9	Immediate wipe signal received	Wiper off	Perform one low-speed wipe cycle