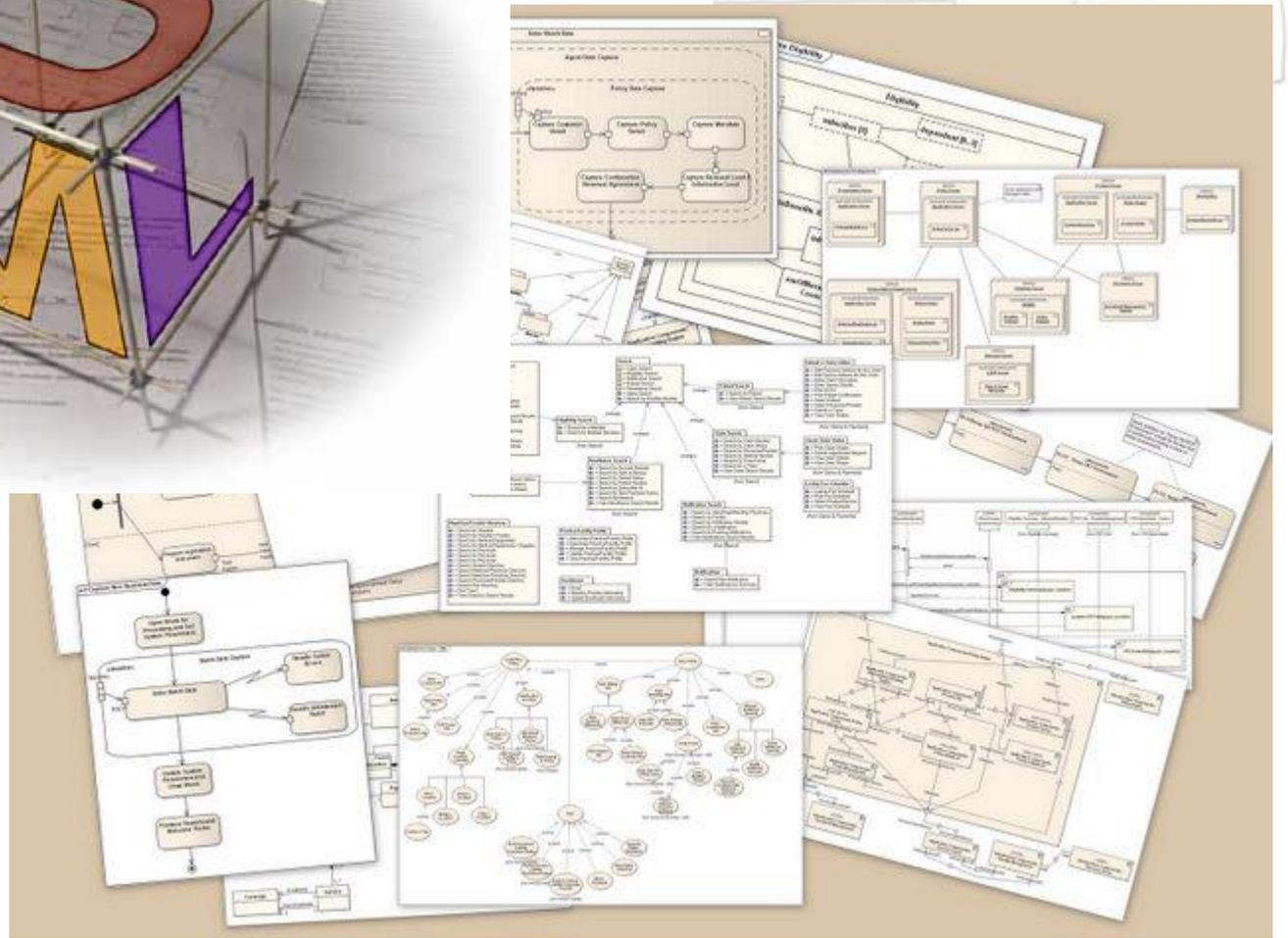
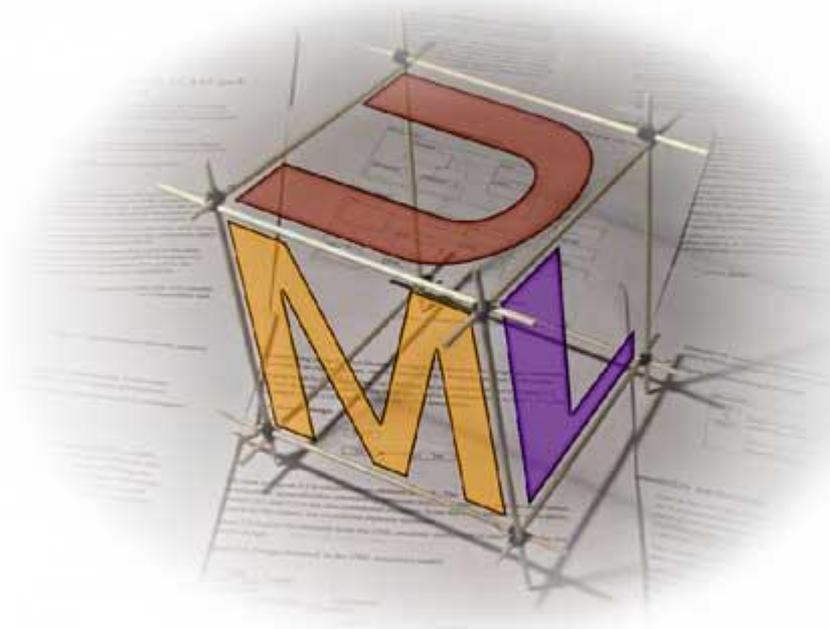


UML e i diagrammi di attività

Sintassi e Linee Guida

Dr. Andrea Baruzzo

andrea.baruzzo@dimi.uniud.it



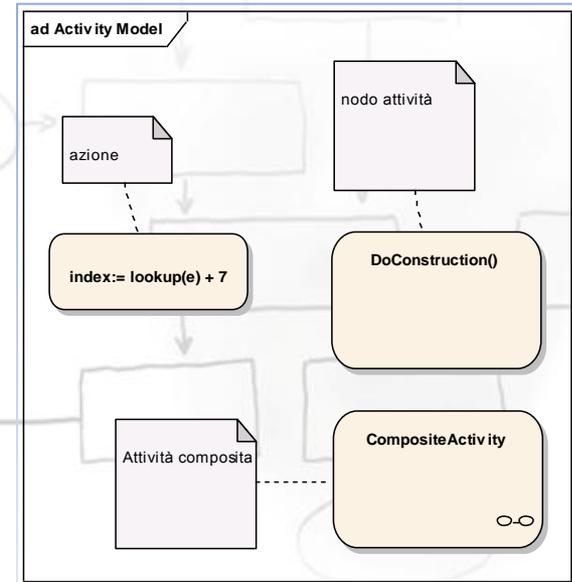
Attività: che cosa sono e a cosa servono

- Un diagramma di attività mostra il **flusso di operazioni** relativo ad un'attività
- Un'attività è un'esecuzione non atomica di operazioni all'interno di una macchina a stati
- L'esecuzione di un'attività viene decomposta in azioni atomiche
- Ogni azione può o meno cambiare lo stato del sistema
- I diagrammi di attività sono spesso usati anche per descrivere la logica di un algoritmo (sono l'equivalente UML dei diagrammi di flusso)
- Graficamente un diagramma di attività è un insieme di archi e nodi (similmente ad un diagramma di stato)

Azioni e nodi attività

- Azioni (atomiche)

- Valutazione di espressioni
- Assegnamenti / Ritorno di un valore
- Invocazione di un'operazione su un oggetto
- Creazione/distruzione di un oggetto

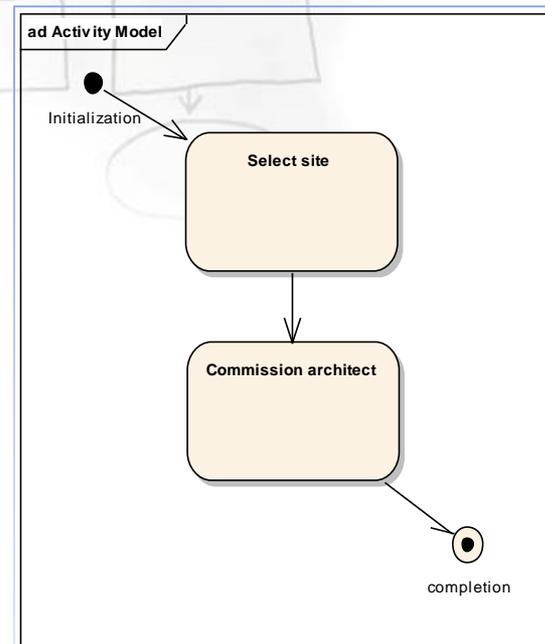


- Nodi Attività

- Raggruppamento di azioni atomiche o di altri nodi attività
- Un'azione può essere vista come un'attività che non può essere ulteriormente decomposta
- Espandendo un nodo attività si ottiene un altro diagramma di attività (attività composta)
- A parte questa differenza, i due concetti sono rappresentati mediante lo stesso simbolo grafico

Flusso di controllo

- Quando un'azione o un'attività
 - viene completata, il flusso di controllo passa al nodo azione (attività) immediatamente successivo
 - Il flusso di controllo viene specificato mediante frecce che collegano due nodi (attività o azione)
- Il flusso mostrato in figura è quello più semplice:
 - il **flusso sequenziale**

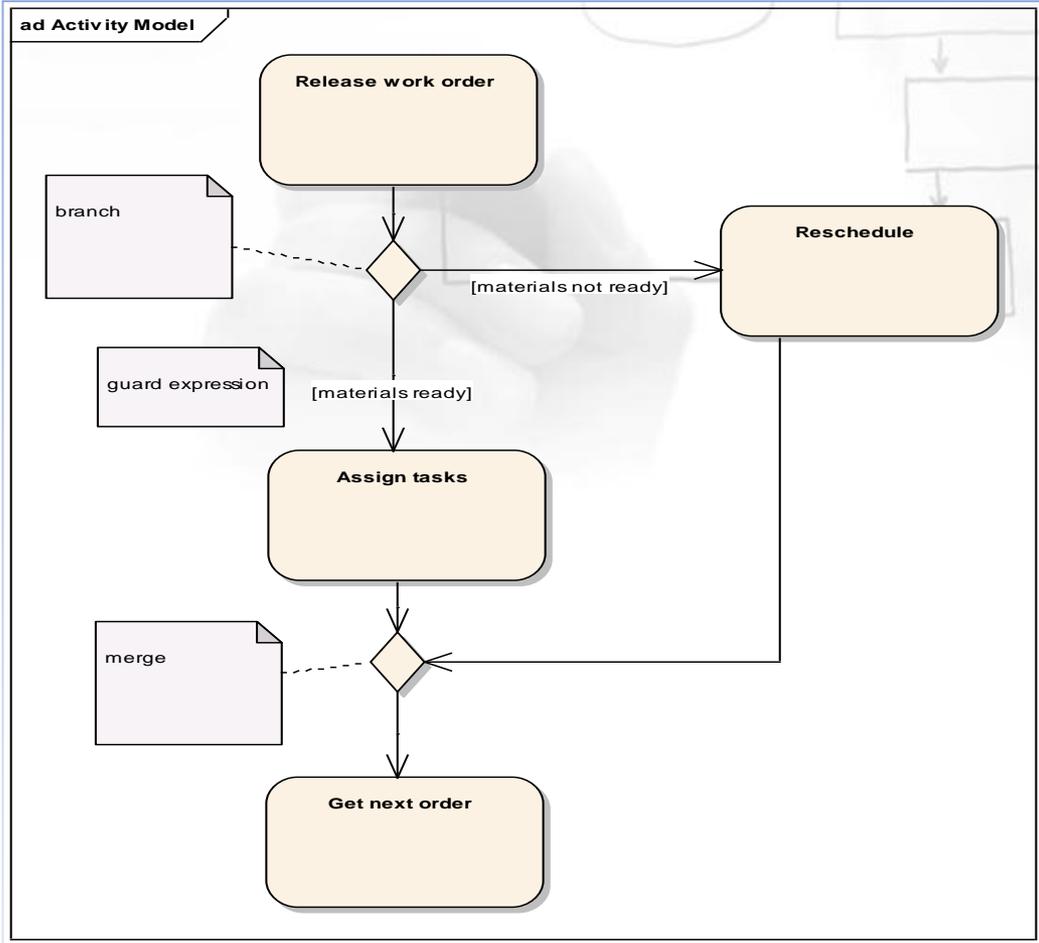


Branch e merge di attività

- Un altro tipo di flusso possibile è il **branch** (diramazione)
- Un branch è rappresentato da un **diamante**
- Ogni branch ha:
 - Un flusso entrante
 - Due o più flussi uscenti
 - Una condizione logica (talvolta implicita) che determina:
 - quale dei flussi uscenti verrà eseguito da una particolare esecuzione
- Quando due flussi si riuniscono, è possibile usare ancora il simbolo di diamante; in questo caso viene detto **merge** (fusione, unione)
- Ogni merge ha almeno due flussi entranti e un flusso uscente



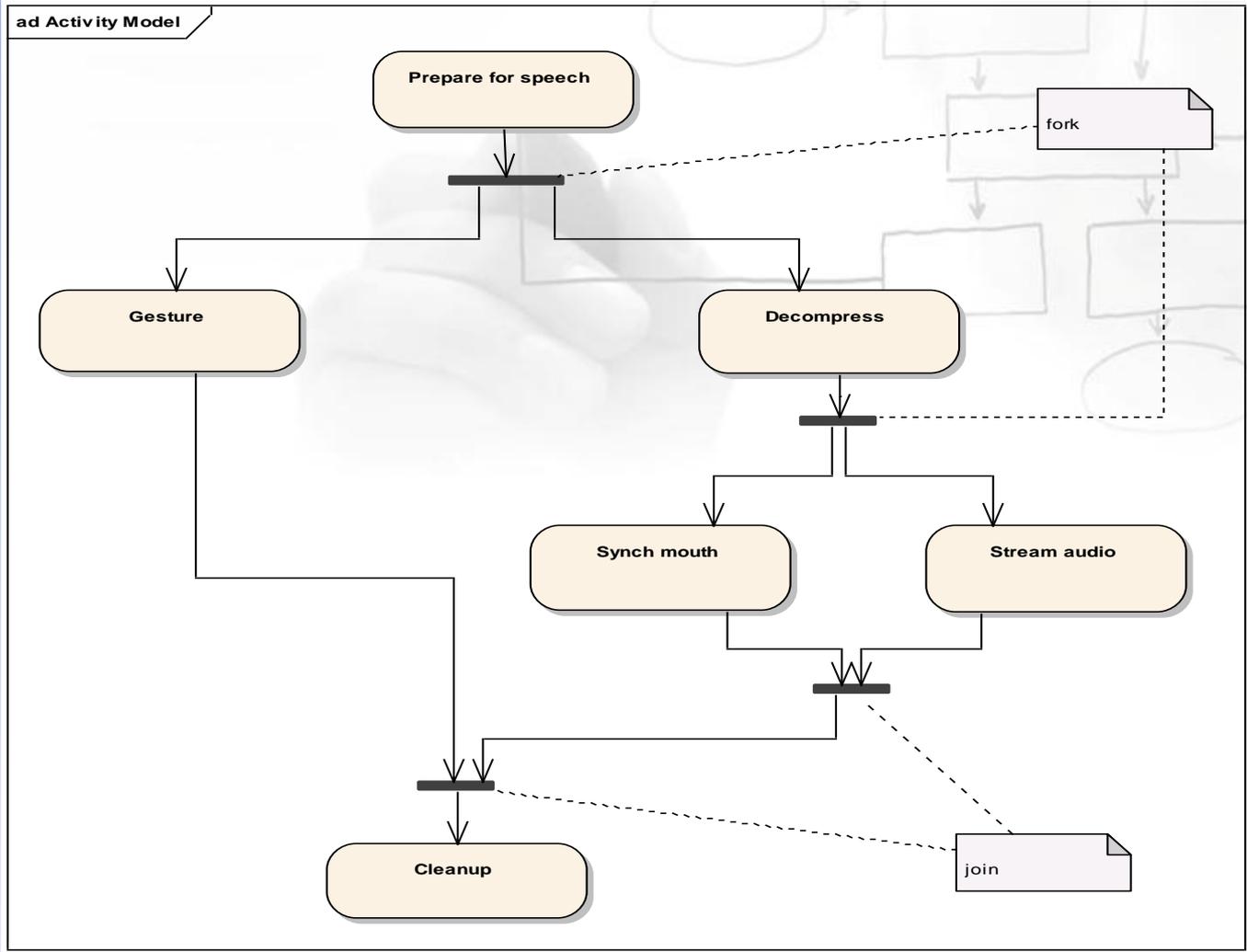
Branch e merge (cont.)



Fork e join

- Alcuni flussi possono essere **concorrenti**
- In UML vengono usate delle **barre di sincronizzazione** per specificare **fork e join** di flussi di controllo paralleli
- Un join rappresenta la sincronizzazione di due o più flussi di controllo concorrenti
 - Un join ha due o più flussi entranti e un flusso uscente
 - La sincronizzazione sul join attende che tutte le attività nei flussi entranti abbiano terminato la loro esecuzione, prima di procedere
- Join e fork si devono bilanciare
- Le attività in un flusso di controllo parallelo comunicano tra loro spedendosi segnali (stile di comunicazione detto coroutine)

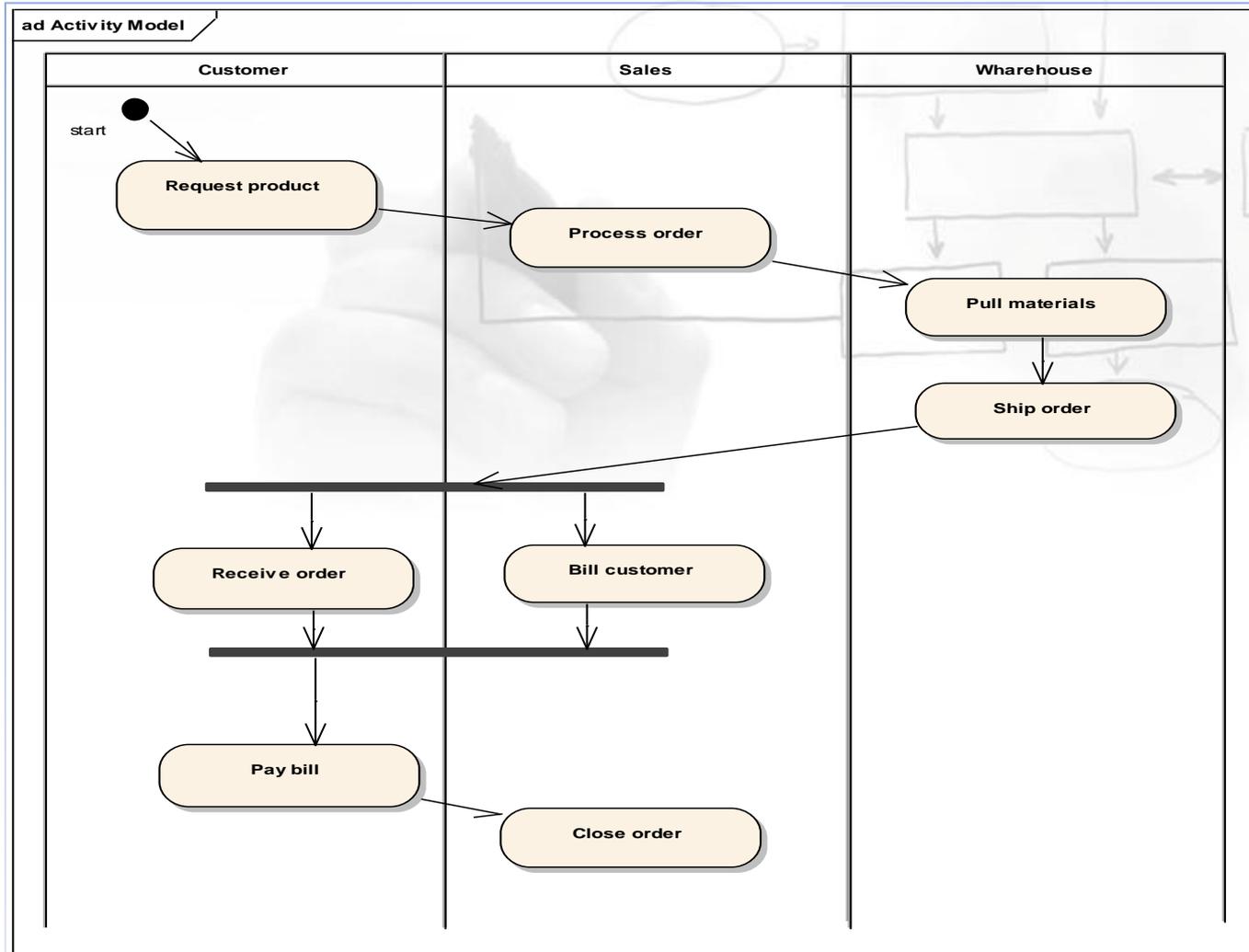
Fork e join (cont.)



Swimlane (piste)

- A volte è utile partizionare le attività in base alle entità coinvolte che le devono svolgere
- In UML si usano a tale scopo delle “piste”, chiamate **swimlane**
- Una swimlane è un **raggruppamento (verticale oppure orizzontale) di attività eseguite da una stessa entità** (ad esempio, una classe)
- Ogni swimlane deve avere un nome univoco nel diagramma
- Le swimlane rappresentano responsabilità/agenti specifiche/specifici nel contesto di un'attività generale
- Le attività sono associate univacamente ad un'unica swimlane
- Solo le transizioni (flussi) possono attraversare due o più swimlane

Swimlane (cont.)



Flusso di oggetti

- A volte è utile evidenziare non solo il flusso di controllo, ma anche gli oggetti coinvolti
- Un'attività può creare un oggetto
- Un'altra attività può contenere azioni che modificano lo stato interno di un oggetto
- Il flusso del valore (stato) di un oggetto tra due azioni è detto flusso dell'oggetto
- Lo stato viene rappresentato tra parentesi quadre all'interno dell'oggetto, oppure come constraint in una nota associata all'oggetto stesso

Flusso di oggetti (cont.)

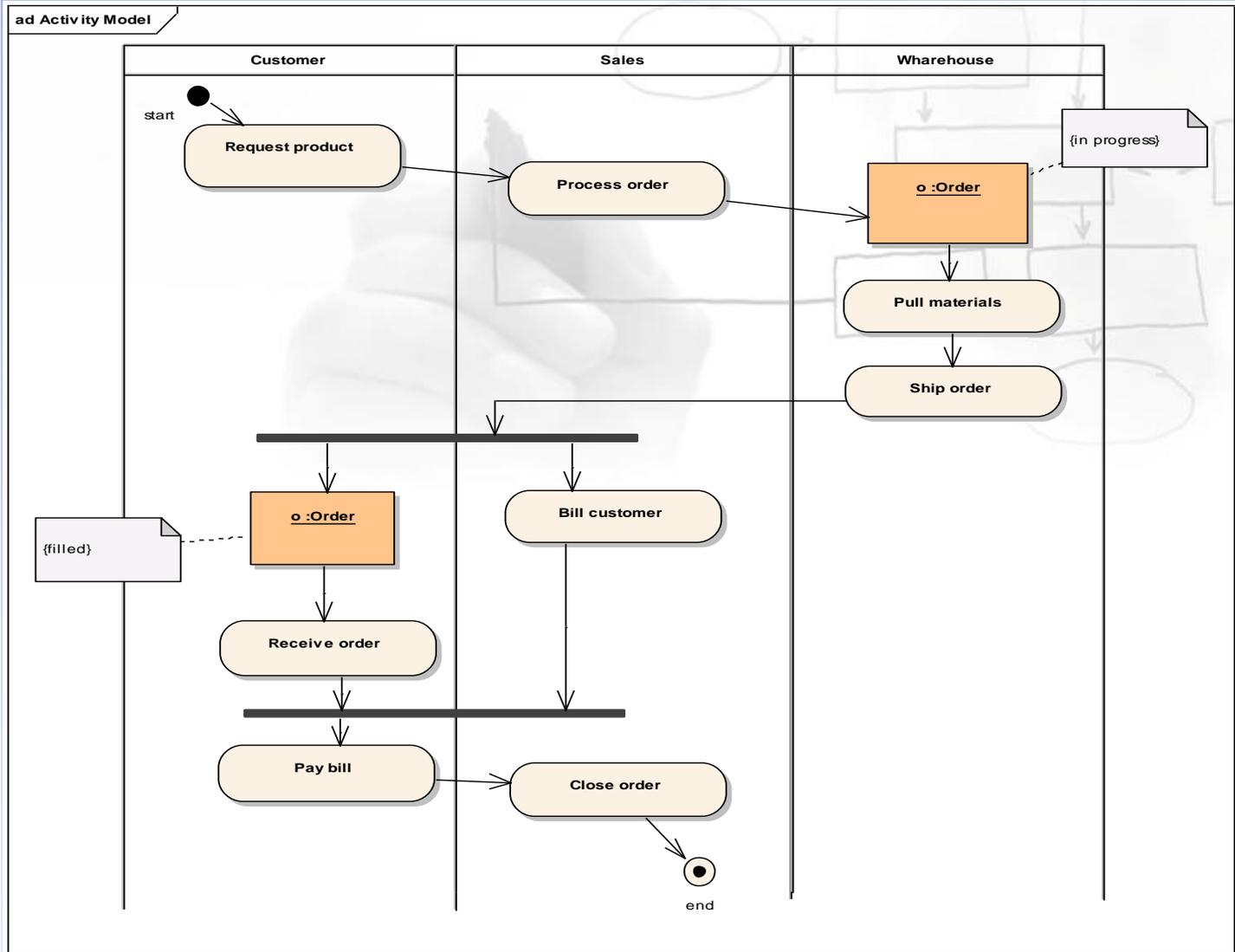


Diagramma di attività e casi d'uso

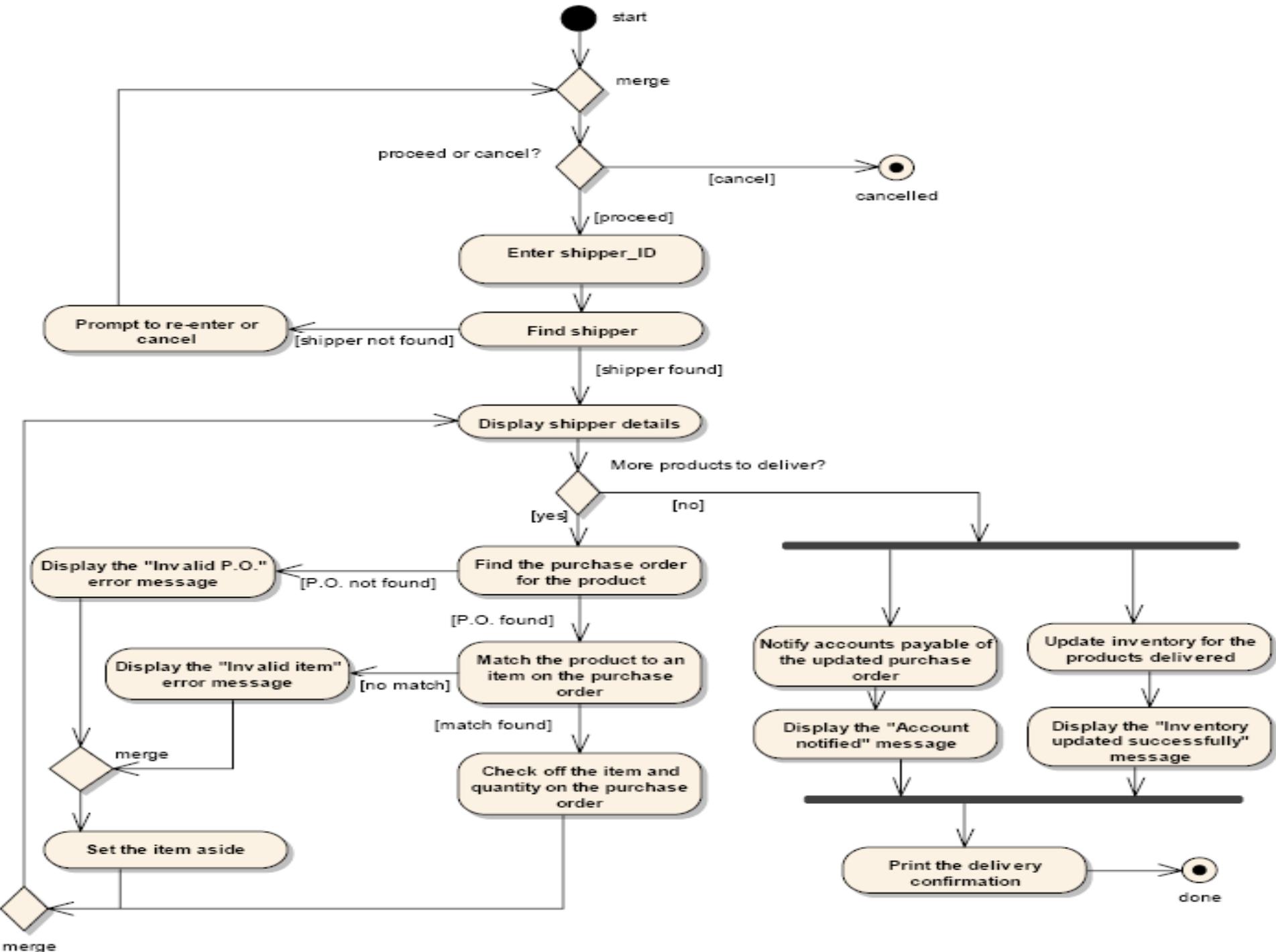
- Un caso d'uso può essere il punto di partenza per la costruzione di un diagramma di attività
- E' sensato: entrambi sono rappresentazioni tipiche dell'analisi di un problema (o di un dominio)
- Il diagramma di attività fornisce una prospettiva algoritmica, mentre i casi d'uso forniscono una prospettiva funzionale
- Le due viste sono correlate, ma non totalmente equivalenti
- Il punto di partenza per costruire un diagramma di attività da un caso d'uso sono le descrizioni testuali, i flussi alternativi, le eccezioni, i singoli passi, le postcondizioni, le condizioni di terminazione
- Vediamo un esempio: la spedizione di un ordine (order shipping)

Diagramma di attività e casi d'uso (cont.)

Descrizione testuale del caso d'uso Spedire Prodotti (Ship Product)	
Nome caso d'uso	Spedire Prodotti (Ship Product)
Scope	Delivery Shipment Subsystem
Goal (summary)	Il magazziniere predispose l'invio dei prodotti di un ordine a mezzo corriere
Attori	Magazziniere
Precondizioni	La spedizione viene effettuata utilizzando un corriere già registrato presso gli archivi del negozio
Descrizione (main success scenario o scenario principale)	<ol style="list-style-type: none">1. Il magazziniere inserisce il codice di identificazione del corriere che si vuole utilizzare per la spedizione;2. Il sistema visualizza le informazioni di riepilogo relative al corriere selezionato;3. Il magazziniere inserisce le informazioni della spedizione (data di spedizione, mittente, destinatario, elenco prodotti e quantità da spedire);4. Per ogni prodotto da spedire:<ol style="list-style-type: none">4.1 Il magazziniere inserisce il numero d'ordine a cui si riferisce il prodotto;4.2 Il sistema cerca una corrispondenza tra l'ordine inserito e il prodotto da spedire;4.3 Il sistema controlla che il prodotto e la quantità richieste nell'ordine corrispondano con i dati di spedizione;

Diagramma di attività e casi d'uso (cont.)

	<p>5 Il magazziniere notifica la bolletta di pagamento relativa all'ordine aggiornato del cliente;</p> <p>6 Il sistema visualizza la notifica relativa alla bolletta di pagamento;</p> <p>7 Parallelamente alle attività ai passi 5 e 6, il magazziniere aggiorna il magazzino per registrare l'uscita dei prodotti inseriti nell'ordine aggiornato che sta evadendo;</p> <p>8 Il sistema visualizza lo stato del magazzino aggiornato;</p> <p>9 Il magazziniere stampa la conferma di spedizione.</p>
Alternative (estensioni)	<p>1a. Se il codice non corrisponde ad alcuna compagnia di corrieri, viene visualizzato il messaggio di errore "Shipper not found";</p> <p>1b. Il sistema chiede al magazziniere se vuole riprovare con un altro codice oppure se preferisce terminare la procedura;</p> <p>1c. Il magazziniere decide di inserire un altro codice;</p> <p>1d. Il caso d'uso riprende dal passo 1 dello scenario principale.</p> <p>4.1a Se l'ordine non viene trovato, viene visualizzato il messaggio di errore "Invalid Product Order";</p> <p>4.1b il prodotto viene scorporato dalla spedizione e viene messo da parte.</p> <p>4.2a Se non viene trovata alcuna corrispondenza tra il prodotto e l'ordine inserito dal magazziniere, viene visualizzato il messaggio di errore "Invalid Item";</p> <p>4.2b Il prodotto viene scorporato dalla spedizione e viene messo da parte.</p>
Postcondizioni	<p>Se la transazione viene completata con successo, il magazzino è aggiornato, viene emessa una bolletta di pagamento relativa all'ordine effettivamente evaso e viene stampata la conferma di spedizione che viene allegata ai documenti della spedizione;</p> <p>Se la transazione viene annullata dal magazziniere, viene ripristinato lo stato del sistema prima dell'inizio del caso d'uso</p>
Condizioni di terminazione	<p>Il caso d'uso termina se:</p> <ul style="list-style-type: none"> • Il sistema visualizza il messaggio di errore "Shipper not found" e il magazziniere decide di annullare la transazione; • Viene emessa la bolletta di pagamento e il magazzino viene aggiornato; • Il magazziniere annulla la transazione



Bibliografia

- [Booch et al., 2005] Grady Booch et al. *“The Unified Modeling Language User Guide 2/E”*, Addison-Wesley, 2005
- [Rumbaugh et al., 2004] J. Rumbaugh et al. *“The Unified Modeling Language Reference Manual 2/E”*, Addison-Wesley, 2004
- [Fowler, 2003] Martin Fowler. *“UML Distilled 3/E”*, Addison-Wesley, 2003
- [Larman, 2004] C. Larman. *“Applying UML and Patterns”*, Addison-Wesley, 2004
- [Pender, 2003] Tom Pender. *“UML Bible”*, Wiley&Sons, 2003



Domande?
Commenti?
Dubbi?