Diagrammi di sequenza di sistema

In teoria, non c'è differenza tra teoria e pratica. Ma, in pratica, c'è.

Jan L.A. van de Snepscheut

Obiettivi

- Identificare eventi di sistema.
- Creare diagrammi di sequenza di sistema per gli scenari dei casi d'uso.

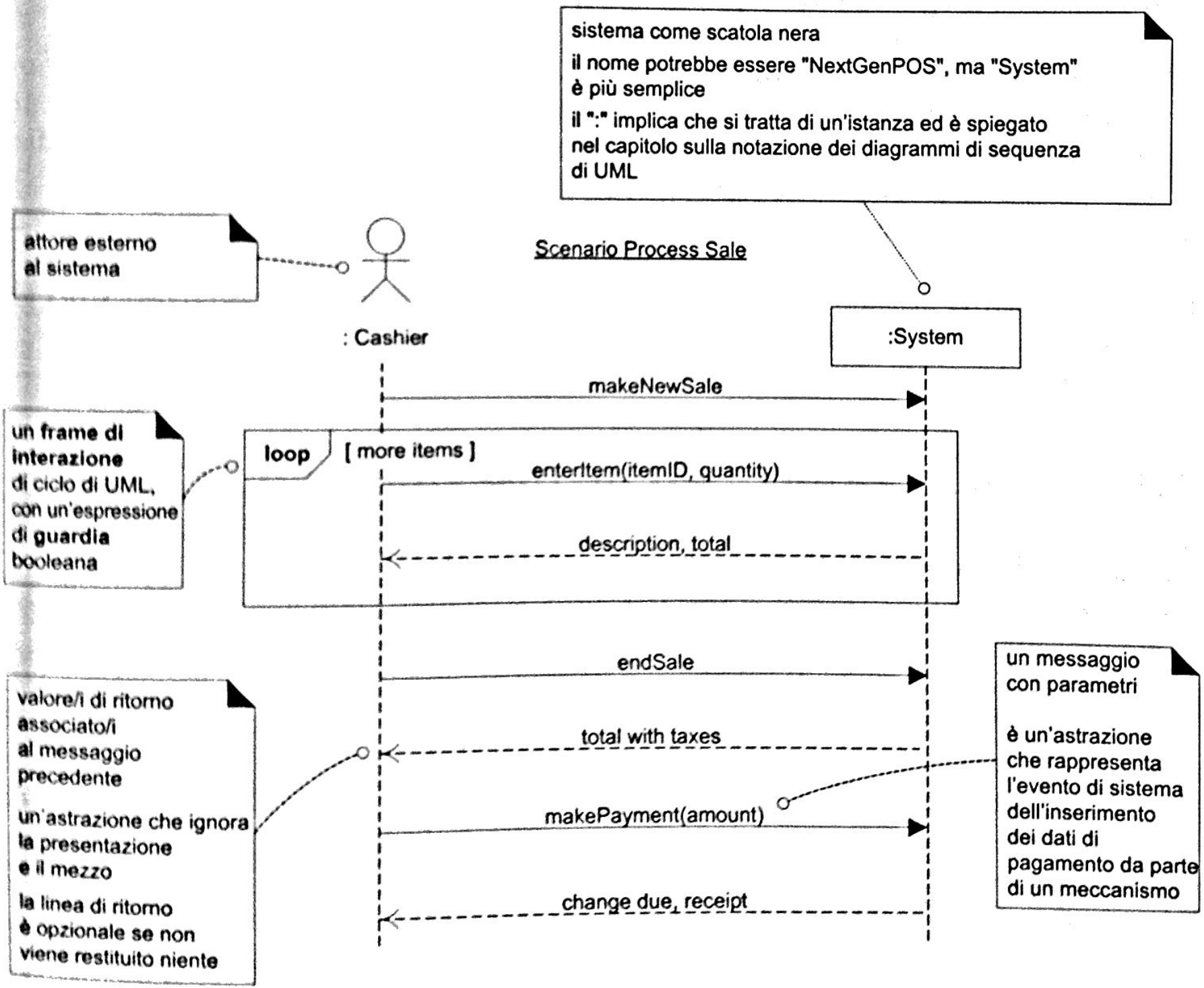

Figura 10.2 SSD per uno scenario di *Process Sale*.

10.2 Che cosa sono i diagrammi di sequenza di sistema

I casi d'uso descrivono il modo in cui gli attori esterni interagiscono con il sistema software che interessa creare. Durante questa interazione un attore genera degli **eventi di sistema** a un sistema, solitamente per richiedere l'esecuzione di alcune **operazioni di sistema**, definite per gestire l'evento. Per esempio, quando un cassiere inserisce l'ID di un articolo, egli richiede al sistema POS di registrare la vendita di quell'articolo (evento *enterItem*). Quell'evento dà inizio a un'operazione sul sistema. Il testo del caso d'uso *implica* l'evento *enterItem*, e il diagramma di sequenza di sistema lo rende concreto ed esplicito.

UML include i **diagrammi di sequenza** come notazione in grado di illustrare interazioni tra attori e le operazioni iniziate da essi.

Un **diagramma di sequenza di sistema** è una figura che mostra, *per un particolare scenario di un caso d'uso*, gli eventi generati da attori esterni, il loro ordine e gli eventi inter-sistema (ovvero, tra sistemi). Tutti i sistemi sono considerati a scatola nera; il diagramma enfatizza gli eventi che superano i confini dei sistemi, dagli attori ai sistemi.

Linea guida

Disegnare un SSD per uno scenario principale di successo di ciascun caso d'uso, e per gli scenari alternativi più frequenti o complessi.

10.3 Motivazione: perché disegnare un SSD

Una domanda interessante e utile nella progettazione del software è la seguente: *Quali eventi solleciteranno il nostro sistema?* È utile e interessante perché il software deve essere progettato per gestire questi eventi (dal mouse, dalla tastiera, da un altro sistema, ...) e generare delle risposte. Fondamentalmente, un sistema software reagisce a tre cose: 1) eventi esterni da parte di attori (umani o sistemi informatici); 2) eventi di timer; e 3) guasti o eccezioni (spesso provenienti da sorgenti esterne).

Pertanto è utile sapere, *con precisione*, quali sono gli eventi esterni di input, ovvero gli **eventi di sistema**. Essi rappresentano una parte importante dell'analisi del comportamento di un sistema.

Il lettore dovrebbe avere familiarità con la nozione di interfaccia di un oggetto software, che descrive l'insieme dei messaggi che possono essere ricevuti dall'oggetto. Questo concetto è utile anche al livello superiore dei componenti, compreso l'intero sistema visto (in modo astratto) come una sola cosa o oggetto.

Prima di passare alla progettazione dettagliata del funzionamento di un'applicazione software, è utile esaminare e definire il suo comportamento a "scatola nera". Il **comportamento di un sistema** è una descrizione di *che cosa* fa un sistema, senza spiegare come lo fa. Un diagramma di sequenza di sistema è una parte di questa descrizione. Altre parti

Un SSD mostra gli eventi di sistema *per un solo scenario di un caso d'uso*, e pertanto è generato per ispezione da un caso d'uso (Figura 10.3).

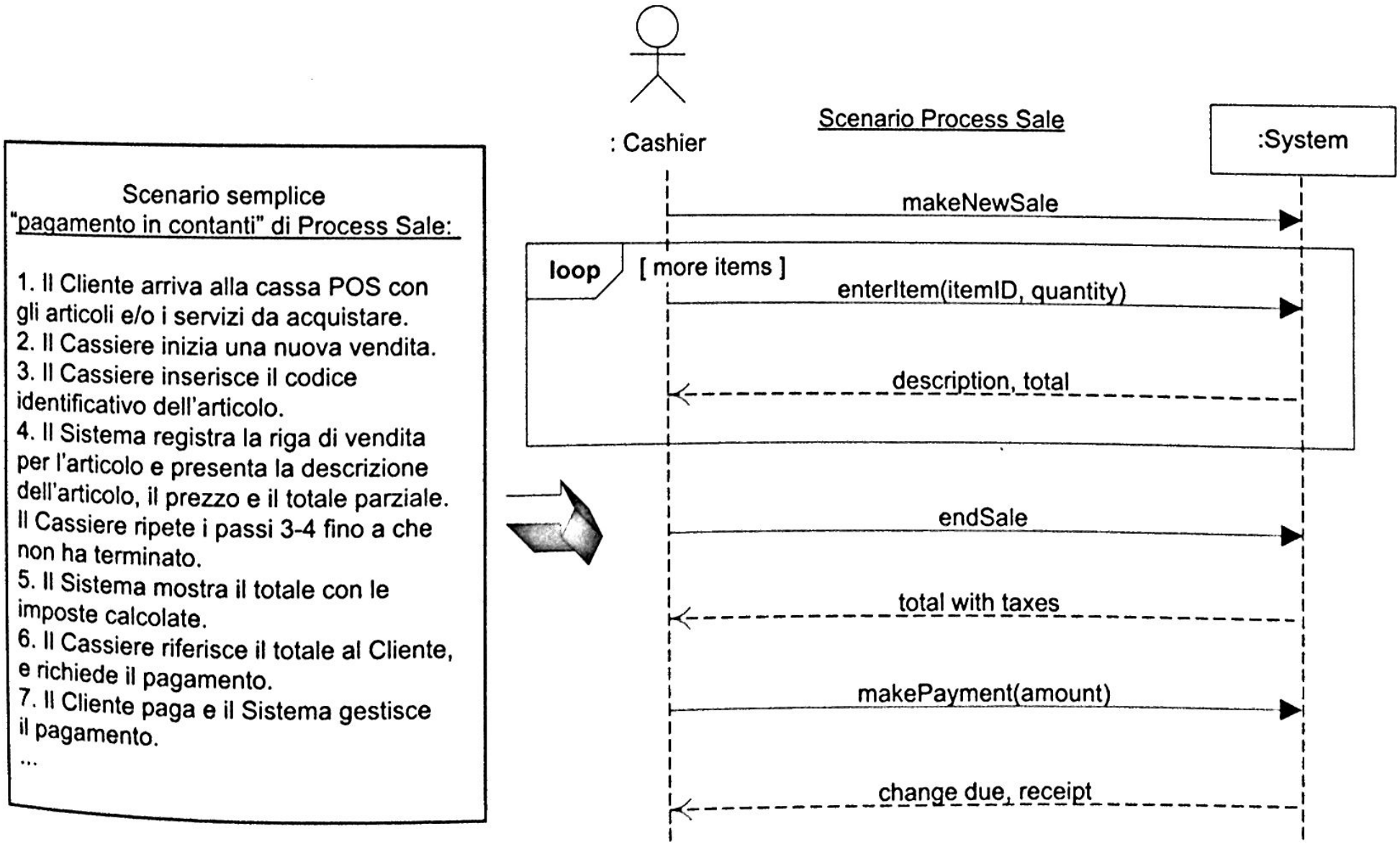

Figura 10.3 Gli SSD derivano dai casi d'uso; un SSD mostra un solo scenario.

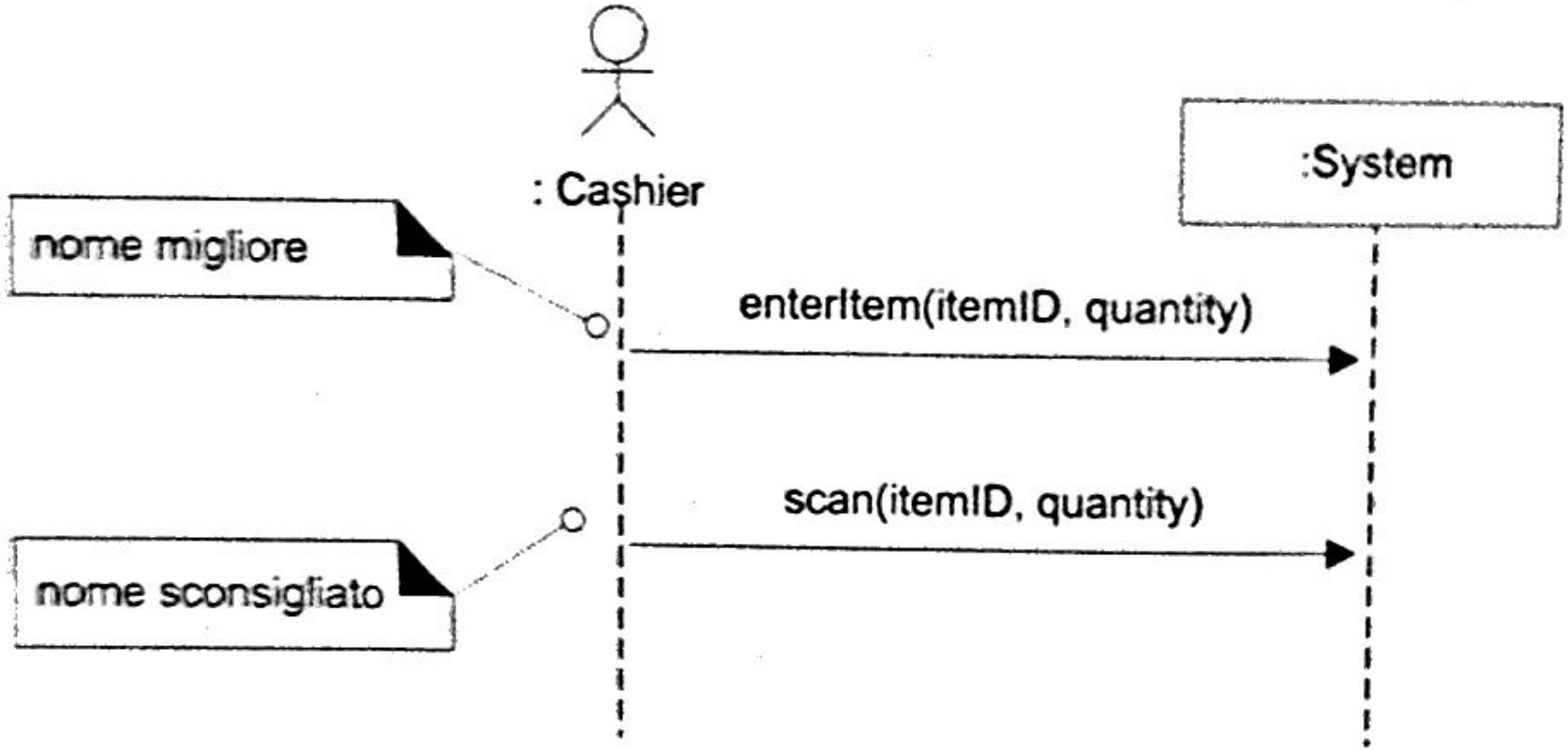

Figura 10.4 Scegliere i nomi degli eventi e delle operazioni a un livello astratto.

Inoltre, è più chiaro iniziare il nome di un evento di sistema con un verbo (add..., enter..., end..., make...), come nella Figura 10.4, poiché sottolinea che si tratta di comandi o richieste.

Linea guida

In generale, è opportuno mostrare nel Glossario i dettagli di molti elaborati.