

## **Statechart diagram for use case**

Adapted from Applying UML and patterns; Craig Larman;

Prentice Hall; 978-0131489066

### **Example: Summary format of a use case**

#### **Process Sale:**

A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and then leaves with the items.

### **Example: Fully dressed use case**

#### **Main Success Scenario: PROCESS SALE**

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total.

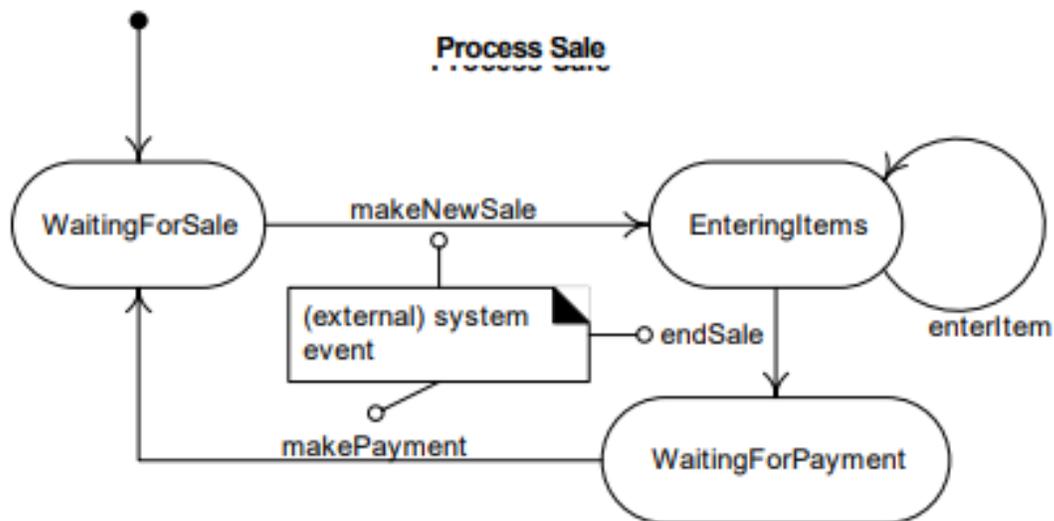
Price calculated from a set of price rules.

Cashier repeats steps 3-4 until indicates done.

5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
8. Customer leaves with receipt and goods (if any).

A useful application of statechart diagrams is to describe the legal sequence of external system events that are recognized and handled by a system in the context of a use case.

A statechart diagram that depicts the overall system events and their sequence within a use case is a kind of use case statechart diagram.



The use case statechart diagram above shows a simplified version of the system events for the Process Sale use case in the POS application.

It illustrates that it is not legal to generate a makePayment event if an endSale event has not previously caused the system to transition to the WaitingForPayment state.

Given a set of use case statechart diagrams, a designer can methodically develop a design that ensures correct system event order.

Extending the Use Case for more functionality

